People's Democratic Republic of Algeria

الجمهورية الجزائرية الديمقراطية الشعبية

Ministry of Higher Education and Scientific Research

وزارة التعليم العالي و البحث العلمي

Kasdi Merbah University of Ouargla

Dissertation Submitted to the Department of Computer Science and Information Technology in Candidacy for the Degree of Master LMD in Industry Artificial Intelligence and Data Science branch
Master degree thesis

Students :

Lina Mohand Oussaid

Hadjer Chekima

Kidney diseases identification using deep and handcrafted models: a comparative study

Evaluation Date: 18/06/2023

Befor the Jury:

Dr.Zerdoumi .O	**President**	**UKM Ouargla**
Dr.Aiadi .O	**Supervisor**	**UKM Ouargla**
Dr.Merabti .H	**Examiner**	**UKM Ouargla**

Dedications

· To our parents.
· To our brothers and sisters.
· To our family and friends.
· To all our teachers from primary school to the end of this thesis.
· To everyone who helped us in completing this work, even if only with advice.

Acknowledgement

This research was undertaken at the Department of Computer Science and Information Technology, Kasdi Merbah Ouargla University. We would like to take this opportunity to express our appreciation. Firstly, we would like to thank God for giving us the strength and patience to carry out this humble work. We are immensely grateful to **Dr. Oussama Aiadi** for his continuous guidance and unwavering support throughout the writing of this thesis. Additionally, we would like to thank our families for their endless support.

Table of content

List of Figures

List of Tables

Abstract

Kidney cysts, tumors, and stones pose significant health risks to individuals, potentially leading to kidney failure if not effectively managed. However, the shortage of nephrologists and kidney specialists globally presents a significant challenge in providing timely and accurate diagnoses.
In this research, we aim to address this challenge by leveraging artificial intelligence (AI) techniques for the detection of kidney cysts, tumors, stones, and normal conditions this was based on early research that made a comparative study between cnn models and transformer variants. To expand that research Our study encompasses an invistigation involving diverse models, including CNN models, transformer variants, lightweight neural networks, and handcrafted features. We evaluate these models on a dataset of 12,446 unique images extracted from abdominal CT scans.
To enhance the comprehensiveness of our investigation, we introduce several modifications to the original dataset, incorporating Gaussian noise, blur, illumination variations (both increase and decrease), and occlusion. These modifications aim to simulate real-world conditions and imaging artifacts, providing a more realistic and challenging dataset for evaluation purposes.
Upon comparing the performance of these models, we observed interesting outcomes. In the original dataset, the BSIF handcrafted features exhibited superior accuracy, achieving an impressive accuracy of 99.51%. In the BLUR dataset, the lightweight neural networks outperformed all other models, achieving an accuracy of 99%. However, in the ILLUMINATION INCREASE dataset, we noted a decline in classifi-cation results when using Vision transformers models (EANet and SWIN). Conversely, in the OCCLUSION dataset, the lightweight neural networks surpassed all other models in terms of accuracy. In the GAUSSIAN dataset, the lightweight neural networks once again outperformed all other models, achieving an accuracy of 99%. Similarly, in the ILLUMI- NATION DECREASE dataset, the lightweight neural networks demonstrated superior accuracy.
Key words:
Deep Learning, Computer Vision, CNN(Convolutional Neural Network), Vision Transformers, Handcrafted features, Lightweight neural networks

Les kystes, les tumeurs et les calculs rénaux présentent des risques importants pour la santé des individus, pouvant entraîner une insuffisance rénale s'ils ne sont pas diagnostiqués et gérés efficacement. Cependant, la pénurie de néphrologues et de spécialistes des reins à l'échelle mondiale constitue un défi majeur pour des diagnostics précis et rapides.

Dans le cadre de cette recherche, nous visons à relever ce défi en exploitant des techniques d'intelligence artificielle (IA) pour la détection des kystes rénaux, des tumeurs, des calculs et des conditions normales. Cette étude s'appuie sur des recherches antérieures ayant réalisé une étude comparative entre les modèles CNN et les variantes des transformateurs. Pour étendre cette recherche, notre étude englobe une investigation portant sur divers modèles, notamment des modèles CNN, des variantes de transformateurs, des réseaux neuronaux légers et des caractéristiques artisanales. Nous évaluons ces modèles sur un ensemble de données comprenant 12 446 images uniques extraites de scanners abdominaux CT. Afin d'améliorer la qualité et l'exhaustivité de notre investigation, nous introduisons plusieurs modifications dans l'ensemble de données d'origine, en incorporant du bruit gaussien, du flou, des variations d'éclairage (augmentation et diminution) et de l'occlusion. Ces modifications visent à simuler des conditions réelles et des artefacts d'imagerie, fournissant ainsi un ensemble de données plus réaliste et plus difficile à évaluer.

En comparant les performances de ces modèles, nous avons observé des résultats intéressants. Dans l'ensemble de données d'origine, les caractéristiques artisanales BSIF ont affiché une précision supérieure, atteignant une impressionnante précision de 99,51 %. Dans l'ensemble de données flou, les réseaux neuronaux légers ont surpassé tous les autres modèles, atteignant une précision de 99 % Cependant, dans l'ensemble de données d'augmentation d'éclairage, nous avons constaté une baisse des résultats de classification lors de l'utilisation des modèles de transformateurs de vision (EANet et SWIN). En revanche, dans l'ensemble de données d'occlusion, les réseaux neuronaux légers ont surpassé tous les autres modèles en termes de

Mots-clés:
Apprentissage profond, Vision par ordinateur, CNN (Réseau de neurones convolutifs), Transformers de vision, Caractéristiques artisanales, Réseaux neuronaux légers.

تشكل الكيسات والأورام والحصى الكلوية مخاطر صحية هامة للأفراد، وقد تؤدي في حال عدم التعامل معها بشكل فعال إلى الفشل الكلوي. ومع ذلك، يشكل نقص أخصائيي الكلى وأطباء الكلوة في جميع أنحاء العالم تحديًا كبيرًا في تقديم تشخيصات سريعة ودقيقة.

في هذا البحث، نهدف إلى معالجة هذا التحدي من خلال استغلال تقنيات الذكاء الاصطناعي (AI) للكشف عن الكيسات الكلوية والأورام والحصى والحالات الطبيعية، وقد اعتمد هذا البحث على أبحاث سابقة قامت بدراسة مقارنة بين نماذج الشبكات العصبية المتكررة والنماذج المتحولة. لتوسيع هذا البحث، تضمنت دراستنا تحقيقًا يشمل نماذج متنوعة، بما في ذلك نماذج الشبكات العصبية المتكررة والنماذج المتحولة وشبكات العصب الخفيفة والميزات المصنوعة يدويًا. قمنا بتقييم هذه النماذج باستخدام مجموعة بيانات تحتوي على ١٢،٤٤٦ صورة فريدة تم استخراجها من فحوصات (CT) البطنية. ولتعزيز شمولية تحقيقنا، قمنا بإدخال عدة تعديلات على مجموعة البيانات الأصلية، بما في ذلك إضافة ضوضاء غاوسية وتشويش وتغييرات في الإضاءة (زيادة وانخفاض) والحجب. تهدف هذه التعديلات إلى محاكاة ظروف واقعية وفنون تصويرية، وتوفير مجموعة بيانات أكثر واقعية و تحديًا لأغراض التقييم.

عند مقارنة أداء هذه النماذج، لاحظنا نتائج مثيرة للاهتمام. في مجموعة البيانات الأصلية، أظهرت الميزات المصنوعة يدويًا من نوع (BSIF) دقة متفوقة، حيث حققت دقة مبهرة تبلغ ٩٩،٥١% . في مجموعة البيانات المشوشة، تفوقت شبكات العصب الخفيفة على جميع النماذج الأخرى، حيث حققت دقة تبلغ ٩٩%. ومع ذلك، في مجموعة البيانات المتعلقة بزيادة الإضاءة، لاحظنا انخفاضًا في نتائج التصنيف عند استخدام نماذج تحول الرؤية ((EANet SWIN)) . وعلى العكس من ذلك، في مجموعة البيانات المتعلقة بالتعتيم ، تفوقت شبكات العصب الخفيفة على جميع النماذج الأخرى من حيث الدقة. في مجموعة البيانات المتعلقة بالضوضاء الغاوسية، قامت شبكات العصب الخفيفة مرة أخرى بتفوق على جميع النماذج الأخرى، حيث حققت دقة تبلغ ٩٩%. بالمثل، في مجموعة البيانات المتعلقة بانخفاض الإضاءة، أظهرت شبكات العصب الخفيفة دقة متفوقة.

الكلمات الرئيسية:
التعلم العميق، رؤية الحاسوب، شبكة عصبية تحويلية ((CNN)) ، التحويلات البصرية، الميزات الحرفية، شبكات عصبية خفيفة الوزن.

General Introduction

Kidney cyst tumors and stones are common conditions affecting the kidneys, and if left untreated, they can lead to serious consequences, including kidney failure.

Kidney failure, occurs when the kidneys lose their ability to function adequately. If kidney cyst tumors and stones are not treated promptly, the damage to the kidneys can progress, ultimately leading to kidney failure.

Timely detection, accurate diagnosis, and appropriate treatment are essential to prevent kidney damage and preserve kidney function. Regular medical check-ups for kidney cyst tumors and stones are crucial steps towards maintaining kidney health and preventing the potential complications that can arise from untreated conditions. However, despite the importance of these interventions, a significant challenge faced globally is the scarcity of nephrologists, medical professionals specialized in kidney diseases and disorders. This scarcity creates a pressing need for innovative solutions to bridge the gap between the increasing demand for nephrological care and the limited availability of specialists.

With the rise of technology and the advent of artificial intelligence (AI), there is an opportunity to address this issue. AI-powered systems have demonstrated promising potential in various areas of healthcare, including radiology and medical imaging. These systems can assist in the detection and analysis of kidney cyst tumors and stones, aiding in early diagnosis and providing valuable insights to healthcare professionals.

AI algorithms can be trained to recognize patterns and abnormalities in medical imaging, enabling them to identify potential kidney cyst tumors and stones with high accuracy. This can help streamline the diagnostic process, allowing healthcare providers to prioritize cases that require immediate attention. Additionally, AI can assist in monitoring the progression of existing cysts and stones, facilitating timely interventions and preventing further complications.

AI-powered tools can act as valuable decision-support systems, assisting healthcare professionals in making informed decisions, interpreting diagnostic results, and providing personalized treatment recommendations.

Collaborative efforts between AI systems and healthcare professionals can lead to improved patient outcomes, enhanced efficiency, and better utilization of limited resources.

Recent studies have demonstrated the potential of transformer-based models and their variants in tackling the challenges posed by kidney cyst tumors and stones. Originally designed for natural language processing tasks, transformers have proven to be effective in diverse fields, including medical imaging and healthcare.

One notable area of research involves comparing transformer-based models with traditional convolutional neural network (CNN) models to evaluate their efficiency in addressing kidney-related conditions. These comparative studies aim to assess the performance of transformers in medical image analysis tasks, such as segmentation, classification, and detection of cyst tumors and stones.

The findings of these comparative studies provide valuable insights into the strengths and limitations of transformers compared to CNN models. While CNN models have been widely used in medical imaging, transformers offer distinct advantages, such as their ability to capture long-range dependencies and model complex relationships within images.

1 A brief overview on the state of the art

The rise of deep learning has led to a significant surge in research focused on its applications, particularly in the areas of image processing and classification. One area that has witnessed substantial growth is the use of deep learning for autodiagnosis of radiological findings and segmentation tasks [9].

The authors in [9] used Vision transformer and explainable transfer learning models for auto detection of kidney cyst, stone and tumor from CT-radiography, swin transformers got the best classification accuracy result of 99.30%.

Aksakalli. et al[15] talked in hispaper about the classification of Kidney x-ray images using both machine learning and deep learning techniques, and evaluated various machine learning methods such as Decision Trees (DT), Random Forest (RF), Support Vector Machines (SVM), Multilayer Perceptron (MLP), K-Nearest Neighbor (kNN), Naive Bayes (BernoulliNB), and deep neural networks using CNN. the Decision Tree Classifier (DT) got the best classification result. This method has the highest F1 score rate with a success rate of 85.3% .

In[16] , pre-trained DNN models such as ResNet-101, ShuffleNet, and MobileNet-v2 are used to extract features from kidney ultrasound pictures, which are then classified using Support Vector Machines (SVM), with final predictions made using the majority voting technique, The presented method resulted in maximum classification accuracy of 96.54% testing with quality images and 95.58% testing with noisy images.

Fu. et al.(2021)[17], used Deep-learning-based CT imaging in the quantitative evaluation of chronic kidney diseases, the RDA-UNET model achieved 96.34% precision, and 96.88% recall for the left kidney and 95.34% precision, and 94.61% recall for the right kidney, which were better than other algorithms.

In their work, Zheng et al. (2019)[18] discussed the computer-aided diagnosis of congenital abnormalities of the kidney and urinary tract. The study involved the integration of texture image features and deep transfer learning image features. The extracted features were then used by the SVM classifier to classify ultrasound images as either normal or abnormal.

The study conducted by Parakh et al. (2019)[19] utilized two consecutive CNN models. The first CNN was designed to identify the urinary tract, and the second CNN was employed to detect the presence of stones. The results obtained from this approach demonstrated a remarkable accuracy of 95%.

Yildirim et al. (2020)[20] proposed an automated method for detecting kidney stones using coronal Computed Tomography (CT) images and employing a deep learning technique, specifically XResNet-50. The study demonstrated impressive results, achieving a detection accuracy of 96.82%.

Zhang et al. (2019) [18] proposed a method that involved incorporating two morphology convolution layers, modifying feature pyramid networks (FPNs) in the faster RCNN framework, and utilizing four thresholds. Their approach achieved an impressive area under the curve (AUC) value of 0.871.

Md. et al.(2022) [21] proposed a segmentation based kidney tumor classification using Deep Neural Network (DNN), finally, the classification models MobileNetV2, VGG16, InceptionV3 scored with 95.29%, 99.21% and 97.38% accuracy on test set.

2 Contributions

Our research in the field of kidney cyst, stone, tumor, and normal condition detection aimed to advance our understanding and capabilities. To achieve this, we conducted an extensive comparative study involving a variety of models, including CNN models, transformer variants, lightweight neural networks, and handcrafted features. In our study, we utilized a dataset comprising 12,446 unique images extracted from abdominal CT scans. These images were carefully categorized into four distinct groups: cyst, normal, stone, and tumor. It is important to note that the dataset included images captured from different sections during the CT scans. The dataset was obtained from Kaggle and collected from Picture Archiving and Communication Systems (PACS) as well as hospital workstations located in Dhaka, Bangladesh. Notably, the images were acquired from patients who had previously been diagnosed with kidney tumor, cyst, normal, or stone conditions. By analyzing this comprehensive dataset and employing various models, we aimed to evaluate the performance of different approaches and contribute to the field of kidney condition detection.

To further expand our study, we introduced several modifications to our original dataset. These modifications included the addition of Gaussian noise, blur, variations in illumination (both increase and decrease), and occlusion. By incorporating these modifications, we aimed to create a more diverse and challenging dataset that would better simulate real-world conditions and potential imaging artifacts. Using this augmented dataset, we conducted a similar study to the original one, evaluating the performance of the different models and approaches in detecting kidney cyst, normal, stone, and tumor conditions. The results obtained from this extended study provided valuable insights into the robustness and generalization capabilities of the models under more challenging conditions.

In our experiment, we conducted evaluations on six different datasets and obtained the following results:

For the blur dataset, the VGG16 model achieved the highest accuracy of 98.08%. This accuracy surpassed the performance of the ResNet50 and Inception-v3 models.Among the transformer variants, the CCT model exhibited the best accuracy of 98.75%. This performance outperformed other transformer models considered in the study.In terms of handcrafted features, both the Local Binary Patterns (LBP) and Histogram of Oriented Gradients (HOG) approaches achieved an accuracy of 99%.Regarding the lightweight neural networks, all of them achieved an accuracy of 99% for the evaluated datasets.

For the gaussian dataset, also VGG16 model achieved the highest accuracy of 94.71%. This accuracy surpassed the performance of the ResNet50 and Inception-v3 models.Among the transformer variants, the CCT model exhibited the best accuracy of 88.94%. This performance outperformed other transformer models considered in the study. In terms of handcrafted features, both BSIF and Histogram of Oriented Gradients (HOG) approaches achieved an accuracy of 99%.Regarding the lightweight neural networks, all of them achieved an accuracy of 99% for the evaluated datasets.

For the Illumination Increase dataset, Resnet-50 model achieved the highest accuracy of 90.29%. This accuracy surpassed the performance of the VGG16 and Inception-v3 models.Among the transformer variants, the CCT model exhibited the best accuracy of 88.65%. This performance outperformed other transformer models considered in the study. In terms of handcrafted features, both BSIF and Histogram of Oriented Gradients (HOG) approaches achieved an accuracy of 99%.Regarding the lightweight neural networks, all of them achieved an accuracy of 99% for the evaluated datasets.

For the Illumination Decrease dataset, VGG16 model achieved the highest accuracy of 96.44%. This accuracy surpassed the performance of the Resnet-50 and Inception-v3 models.Among the transformer variants, the CCT model exhibited the best accuracy of 97.69%. This performance outperformed other transformer models considered in the study. In terms of handcrafted features, both BSIF and Histogram of Oriented Gradients (HOG) approaches achieved an accuracy of 98.84% and 99%.Regarding the lightweight neural networks, all of them achieved an accuracy of 99% for the evaluated datasets.

For the Original dataset, VGG16 model achieved the highest accuracy of 90.38%. This accuracy surpassed the performance of the Resnet-50 and Inception-v3 models.Among the transformer variants, the CCT model exhibited the best accuracy of 96.73%. This performance outperformed other transformer models considered in the study. In terms of handcrafted features, both BSIF and Histogram of Oriented Gradients (HOG) approaches achieved an accuracy of 99%.Finally in terms of lightweight neural networks MDFNet got the best accuracy of 99%

For the Occlusion dataset, VGG16 model achieved the highest accuracy of 93.46%. This accuracy surpassed the performance of the Resnet-50 and Inception-v3 models.Among the transformer variants, the CCT model exhibited the best accuracy of 93.37%. This performance outperformed other transformer models considered in the study. In terms of handcrafted features, Histogram of Oriented Gradients (HOG) achieved an accuracy of 99%.Finally in terms of lightweight neural networks DCTnet and MDFNet got the best accuracy of 99%

Chapter 1

Background

1.1 Introduction

Deep learning has become a popular tool in fields such as computer vision, speech analysis, and natural language processing, where large amounts of data need to be analyzed and human-like intelligence is required. It is also becoming increasingly important in the field of medical image analysis, as evidenced by a recent special issue on the topic. Medical imaging has long been a diagnostic method in clinical practice, and recent advancements in hardware, safety procedures, computational resources, and data storage have greatly benefited this field. Medical image analysis currently focuses on segmentation, classification, and abnormality detection using images generated from a variety of clinical imaging modalities. MIT's technological review has recognized deep learning as one of the top ten breakthroughs of 2013. This chapter will discuss the technologies that we have used throughout this project, their application to medical analysis, and previous work related to our project[22].

1.2 Deep Learning

Deep Learning (DL) is a subset of machine learning that involves training artificial neural networks to learn patterns and relationships in large datasets.
The term deep refers to the multiple layers of neurons that make up the network. It is especially useful for domains such as computer vision, speech analysis, and natural language processing (NLP), where large volumes of data need to be processed and human-like intelligence is required.
DL is inspired by the information processing patterns found in the human brain and is capable of accurately and efficiently assigning credit across numerous layers of a neural network without supervision.
Each neuron in a network receives input from other neurons and performs a simple calculation before passing on the result to other neurons in the next layer. By stacking layers of neurons, deep neural networks can learn to represent complex patterns in data. The training process for a deep neural network involves presenting it with labeled data and adjusting the weights and biases of the neurons so that the network can accurately predict the correct labels. This is done using a technique called backpropagation, which involves computing the gradient of the error with respect to the weights and biases and adjusting them in the opposite direction of the gradient. One of the main advantages

of deep learning is its ability to automatically learn features from raw data, which can significantly reduce the amount of preprocessing required.

DL outperforms traditional ML methods, especially for complex non-linear process models, and has gained popularity for simplifying the improvement of various learning fields, including image recognition and object detection.
DL is also becoming increasingly important in medical image analysis, with recent advances in hardware design, safety procedures, computational resources, and data storage capabilities greatly benefiting the field [23].

1.3 Artificial Neural Networks

Neural networks draw their inspiration from the intricate information processing mechanisms found in biological nervous systems, such as the human brain. The foundation of neural networks can be traced back to the pioneering work of McCulloch and Pitts in 1943[24]. In their influential paper, they proposed a simple mathematical model for a single neuron, which is depicted in Figure 1.1. This model served as a starting point for the development of more sophisticated neural network architectures.[1]

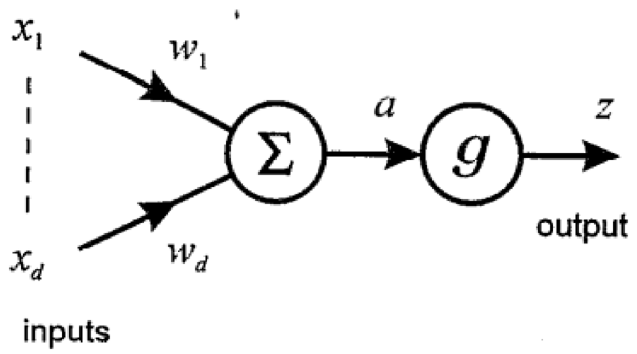


Figure 1.1: The McCulloch-Pitts model of a single neuron [1].

The McCulloch-Pitts model of a single neuron involves combining weighted inputs, denoted as $x_1, \ldots, x_d$, to form a weighted sum represented by Eq. 1.1 . This weighted sum is then passed through a non-linear activation function $g(\cdot)$ as shown in Eq. 1.3, resulting in a final output denoted as z. This activation function introduces non-linearity to the neuron's response, allowing it to capture more complex patterns and relationships in the input data.[1]

$$a = \sum_{i=1}^{d} w_i x_i + w_0 \tag{1.1}$$

In the McCulloch-Pitts model, an additional parameter called the bias, denoted as w_0, is introduced. The bias acts as an offset and can be viewed as a special case of a weight assigned to an extra input x_0, which is constantly set to a value of +1. This allows us to express the weighted sum equation, represented by Eq. 1.1, in the following form [1]:

$$a = \sum_{i=0}^{d} w_i x_i \tag{1.2}$$

where $x_0 = 1$. Note that the weights (and the *bias*) can be of either sign, corresponding to excitatory or inhibitory synapses. The output z of the unit (which may loosely be regarded as analogous to the average tiring rate of a neuron) is then given by operating on a with a non-linear activation function $g(\cdot)$ so that

In the McCulloch-Pitts model, an additional input x_0 is set to a constant value of 1. It is important to note that the weights, including the bias, can have positive or negative values, representing excitatory or inhibitory synapses respectively. The unit's output z, is obtained by applying a non-linear activation function $g(\cdot)$ to the weighted sum a. This process can be described as follows [1]:

$$z = g(a) \tag{1.3}$$

Some possible forms for the function $g(\cdot)$ are shown in Figure 1.2.[1]

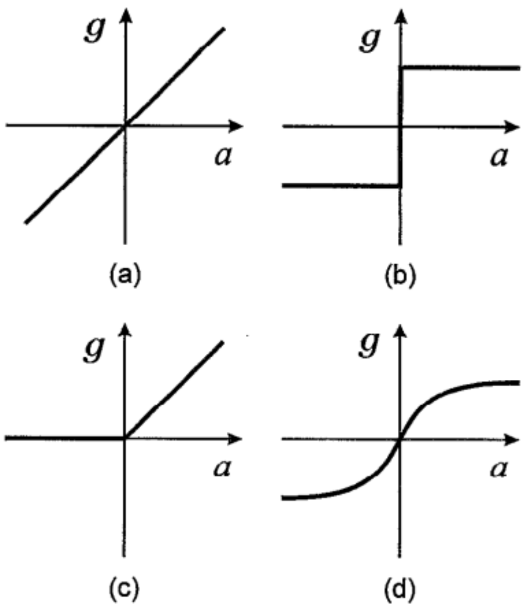


Figure 1.2: A selection of typical activation functions: (a) linear, (b) threshold, (c) threshold linear, (d) sigmoidal. The multilayer perceptron network makes use of sigmoidal units to give network mapping functions which are both non-linear and differentiable [1].

A neural network involves connecting multiple perceptrons together to create a more powerful and flexible computational model, it consists of multiple layers of interconnected neurons, where each neuron performs a similar computation as a perceptron.

In a neural network, the neurons are organized into layers: an input layer, one or more hidden layers, and an output layer. The input layer receives the initial input data, and the output layer produces the final output. The hidden layers, located between the input and output layers, play a crucial role in learning and extracting complex features from the input data. Each neuron in a neural network receives inputs from the neurons in the previous layer. The connections between neurons in a neural network have associated weights, similar to the perceptron model. These weights determine the strength of the connections and are adjusted during the learning process to optimize the network's performance. The learning process, often referred to as training, involves feeding the network with labeled training data and updating the weights based on the errors between the predicted outputs and the true outputs. By combining multiple neurons and organizing them into layers, neural networks can learn to solve complex problems, such as pattern recognition, classification, regression, and sequence generation. The architecture and configuration of a neural network can vary depending on the specific problem domain and desired performance.

1.4 Medical Image Analysis

Medical imaging is a crucial component of modern healthcare, providing visual information about the human body. Its main purpose is to assist radiologists and clinicians in improving the efficiency of diagnostic and treatment processes. Medical imaging encompasses various techniques and technologies that enable the visualization of different body structures and functions. These include X-ray, computed tomography (CT), magnetic resonance imaging (MRI), positron emission tomography (PET), ultrasound, and hybrid modalities. These imaging modalities play a vital role in detecting anatomical and functional details of organs, aiding in both diagnosis and research. Figure 1.3 illustrates a range of common medical imaging modalities used for different body parts, obtained through radiology and laboratory settings. By providing essential insights, medical imaging plays an indispensable role in modern healthcare systems [2].

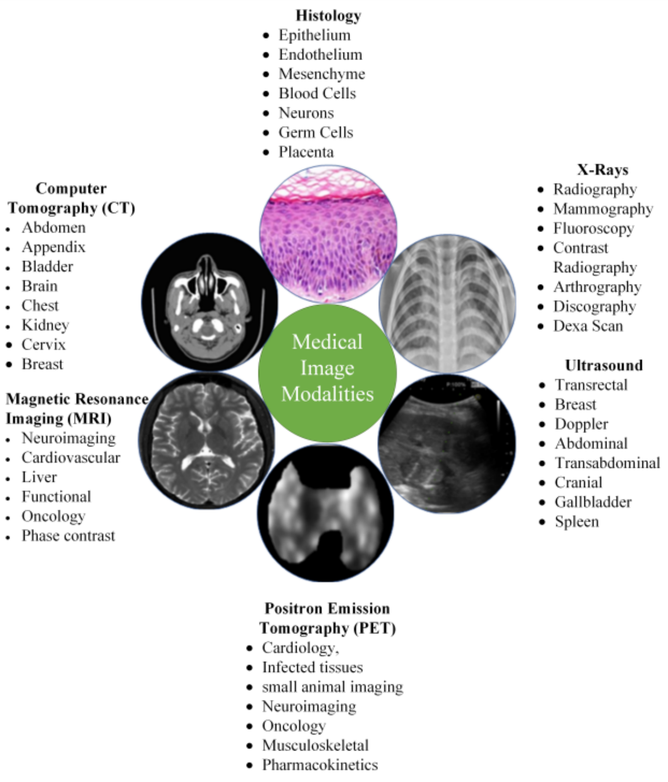


Figure 1.3: Typology of medical imaging modalities [2].

1.5 Computer Vision

Vision is a prime illustration of a task that humans excel at but poses challenges for machines. The human eye effortlessly absorbs an enormous amount of visual information, which the brain processes seamlessly and unconsciously. In contrast, computer vision aims to replicate and, if feasible, enhance this remarkable ability in computers. From a technical perspective, computer vision can be defined as the scientific field dedicated to enabling computers to acquire, process, and analyze digital images. An example of a computer vision application is depicted in Figure 1.4 [3].

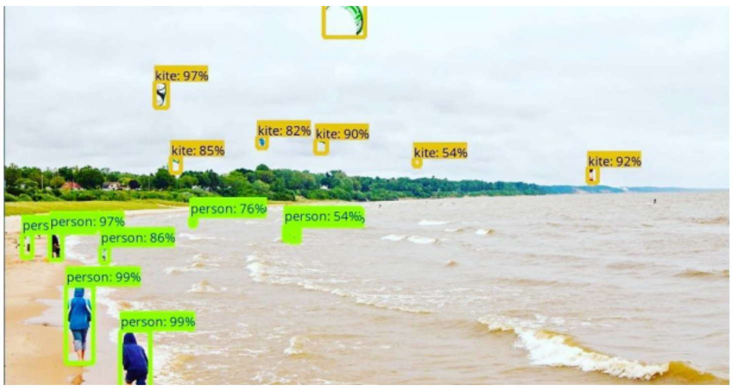


Figure 1.4: Object detection in computer vision. [3].

1.6 Deep Learning in Medical Imaging

1 Diabetic Retinopathy (DR)

Diabetic Retinopathy (DR) is an eye disease that occurs as a complication of diabetes and can lead to vision loss over time. Early detection of DR through retinal screening tests is crucial for effective management and treatment. However, the manual detection of DR is a challenging and time-consuming process due to limited equipment and expertise. On the other hand, the utilization of deep learning models for automated DR detection has shown improved accuracy and optimization, with reported accuracies of up to 94.96%, and [25], with an accuracy of 75%.

2 Cardiac Imaging

Deep learning has provided extremely promising result for cardiac imaging especially for Calcium score quantification. Number of diverse applications has been developed, CT and MRI are the most used imaging modality whereas common task for image segmentation are left ventricle. Manual identification of CAC in cardiac CT requires substantial expert interaction, which makes it time-consuming and in- feasible for large-scale or epidemiological studies. To overcome these limitations, (semi)-automatic calcium scoring methods have been proposed for CSCT [26].

3 Alzheimer's and Parkinsons Diseases Detection

Parkinson's disease (PD) is a neurological disorder characterized by a gradual loss of motor control and impaired sensorimotor integration, likely originating from dysfunction in the basal ganglia. PD is linked to the degeneration or deterioration of dopaminergic neurons. Diagnostic procedures for AD typically involve neurological tests, including brain scans and other assessments. In a study, researchers employed a convolutional neural network (CNN) model called LeNet-5, utilizing scale and shift invariant features such as data shape, mean, and standard deviation. The study achieved an accuracy of 96.86% in detecting brains affected by Alzheimer's disease.

1.7 Conclusion

In conclusion, this chapter has explored the intersection of deep learning, artificial neural networks, medical image analysis, and computer vision in the context of diagnosing and analyzing medical conditions. Throughout the chapter, we have discussed the significance and potential of these fields in advancing healthcare and improving patient outcomes.

Chapter 2

Proposed Methods

2.1 Introduction

The global concern of renal failure and the limited availability of nephrologists worldwide have necessitated the development of an AI-based system for automated kidney disease diagnosis. Early detection of kidney disorders such as cysts, stones, and tumors is crucial in preventing kidney failure. Traditionally, the identification of kidney diseases heavily relied on manual assessment by nephrologists and radiologists, whose numbers are significantly limited globally [9]. To overcome these challenges, advanced technologies leveraging artificial intelligence, particularly deep learning methods, Transformers, Lightweight neural networks, and Handcrafted features have emerged as effective tools for recognizing various kidney diseases.

Convolutional neural network (CNN) has gained significant popularity as a powerful deep learning method, particularly in object classification tasks, building upon the foundations of multilayer perceptron (MLP) [27]. CNN, with its deep structure and incorporation of convolutional computation, simulates the mechanisms of biological vision, automatically extracting relevant features from images. Its ability to perform complex feature extraction contributes to its success in image identification tasks [28]. Transformers, originally introduced for natural language processing tasks, have shown great potential in image analysis as well. These models utilize self-attention mechanisms to capture global dependencies within the input data, enabling them to effectively process and analyze complex medical images.[7] Lightweight neural networks, on the other hand, focus on reducing the computational complexity and memory requirements of traditional deep learning models. These networks are designed to be more efficient, making them suitable for deployment on resource-constrained devices or in real-time applications.[**?**] In addition to deep learning models, handcrafted features have been widely used in medical image analysis. These features are manually designed and extracted from images using various techniques[11, 29, 12, 30, 31].

In this section, we provide an overview of the dataset used in our research, the CNN models employed in our study, with a specific focus on transfer learning techniques. Additionally, we delve into transformer variants, lightweight neural networks, and handcrafted features utilized in our research. Each of these approaches plays a pivotal role in addressing the research objectives of our study. By delving into the intricacies of these models and techniques, we aim to demonstrate their effectiveness and potential in advancing the field of kidney disease analysis.

2.2 Data Collection

This section covers the process of collecting data, as well as the techniques employed for data pre-processing. Additionally, it explores the various modifications applied to the images to create new datasets incorporating these altered versions.

1 Dataset description

The dataset utilized in this project ”CT KIDNEY DATASET: Normal-Cyst-Tumor and Stone” consists of 12,446 distinct images of abdomen CT scans. These images are categorized into four groups: cyst, normal, stone, and tumor Table 2.1. Importantly, the dataset includes images representing various sections generated by a CT scan Figure 2.1. The dataset was sourced from *Kaggle* and collected from PACS and hospital workstations in Dhaka, Bangladesh. These images were obtained from patients who had already been diagnosed with kidney tumor, cyst, normal, or stone findings.

Table 2.1: Number of images of each class of kidney Cyst, Normal, Stone and Tumor dataset.

kidney dataset	Cyst	Normal	Stone	Tumor
2,446	3,709	5,077	1,377	2,283

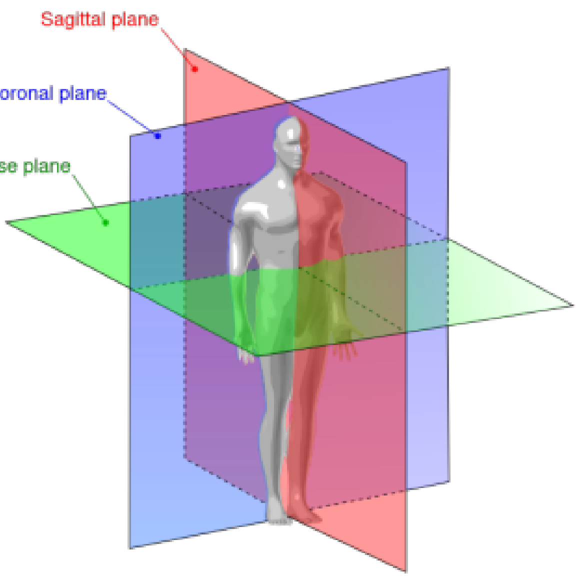


Figure 2.1: CT Scan Cuts [3].

We employed the *cv*2 package to apply modifications to the original images, enabling us to create new datasets that incorporate various transformations. These transformations included blurring, Gaussian noise, changes in illumination (both increase and decrease), and occlusion. Figure 2.2 showcases a sample selection from our diverse range of datasets. In our study We used TensorFlow as our model development library for many reasons. First, TensorFlow provides excellent support for NVIDIA GPUs, which boosts the performance of our models, and second, integrating Keras within TensorFlow is extremely important to us. Keras is a high-level API that takes advantage of TensorFlow's vast capabilities, enabling us to efficiently build machine and deep learning solutions.

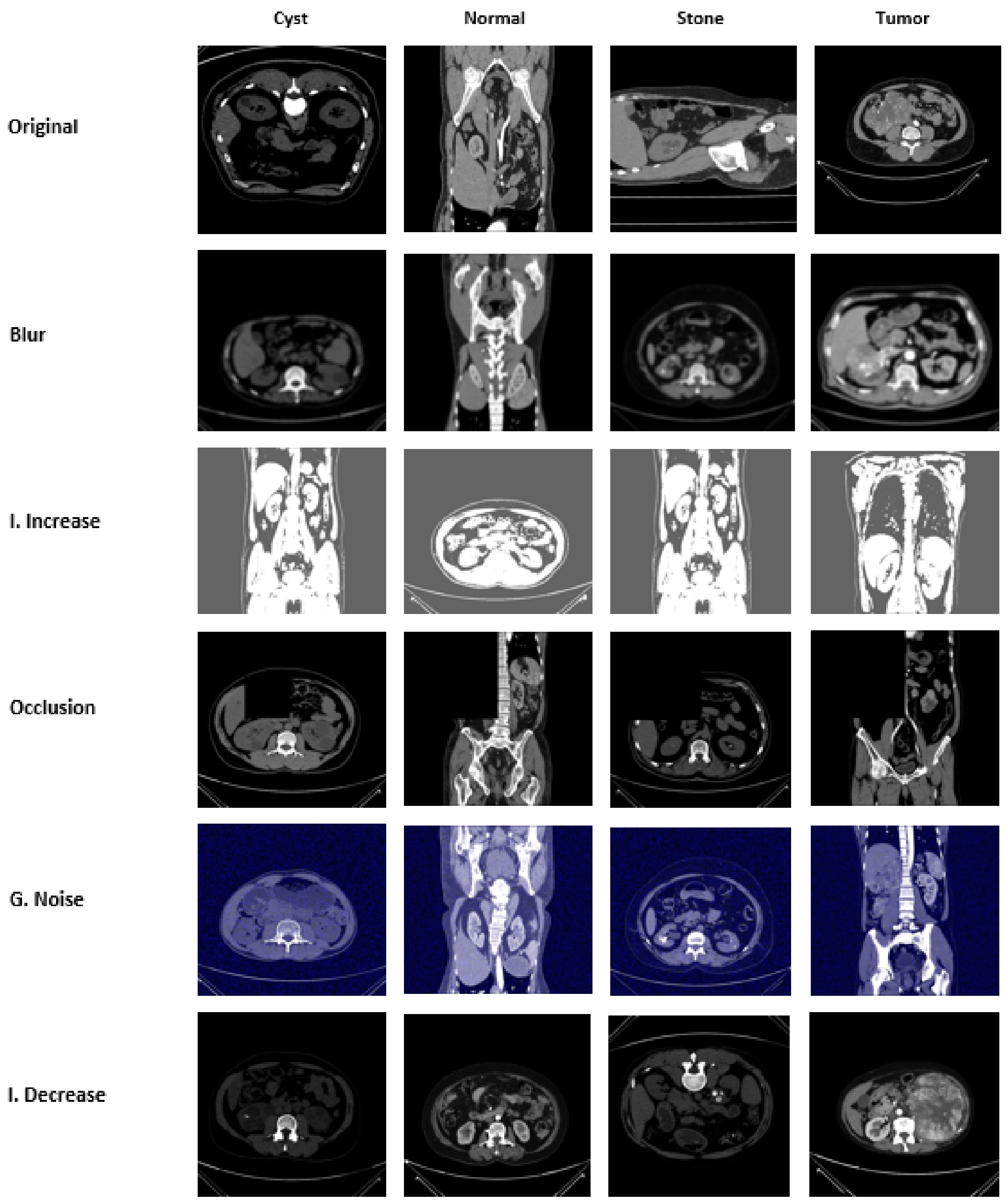


Figure 2.2: a selection from diverse range of datasets (Original, Blur, Illumination increase, Occlusion, Gaussian Noise, Illumination Decrease).

2.3 Deep transfer Learning

Unlike general natural image recognition tasks, medical image analysis lacks large labelled training datasets. Transfer learning involves training a machine learning algorithm on a partiallyrelated or un-related dataset, as well as a labelled training dataset, to circumvent the obstacle of insufficient training data. Essentially the weights learned or pre-trained during the training of a CNN on one (partiallyrelated or un-related) dataset are transferred to a second CNN, which is then trained on labelled medical data using these weights. The weights can be applied to some or all layers of the CNN, except the last fully connected layer.

Transfer learning is a powerful technique in deep learning that enables the effective utilization of previously learned knowledge to solve new tasks with minimal training or fine-tuning. Unlike traditional machine learning methods, deep learning requires a large amount of labeled training data, which can be challenging and costly to obtain, especially in domains like the medical sector. Additionally, deep learning models often demand significant computational resources. To address these issues, Deep Transfer Learning (DTL) has emerged as a valuable approach. DTL involves a two-stage process: pre-training and fine-tuning. In pre-training, a model is trained on a source task, and then in fine-tuning, the model is further trained on the target task. This transfer of knowledge from the pre-trained model to the new model allows for effective training even with limited data [32].

Transfer learning is a highly effective technique in the field of deep learning, enabling the utilization of previously acquired knowledge to solve new tasks with minimal training or fine-tuning. In contrast to traditional machine learning approaches, deep learning requires a substantial amount of labeled training data, which can be challenging and costly to obtain, particularly in domains such as the medical field. Moreover, deep learning models often impose significant computational demands. To address these challenges, Deep Transfer Learning (DTL) has emerged as a valuable solution. DTL involves a two-stage process: pre-training and fine-tuning. During the pre-training stage, a model is trained on a source task, and subsequently, in the fine-tuning stage, the model is further trained on the target task. This transfer of knowledge from the pre-trained model to the new model facilitates effective training even in scenarios with limited available data [32].

2.4 Classification using VGG-16

In this study, we used VGG16 which was has been trained previously in ImageNet dataset as the model. [27] VGG16 is a convolutional neural network model developed by the Visual Geometry Group (VGG) of the University of Oxford and the winner of the 2014 ILSVRC object identifcation algorithm23.[28]
The critical work of VGG16 is to demonstrate that extending the depth of the network can improve the performance of the network in certain situations. Compared with the classic AlexNet, VGG16's improvement lies in the use of multiple 3×3 convolution cores to replace the larger convolution cores (11×11, 7×7, 5×5), which can broaden the depth of the network to improve the network performance efectively, and the use of smaller convolution cores can also reduce the number of network parameters. [28]
The VGG16 network model comprises 13 convolutional layers, three fully connected layers and five pooling layers.
In our experiment, the 16-layer VGG 16 model was tweaked in the last few layers by using

the last layer of the original VGG16 model, and we added average pooling, fattening, and a dense layer with a relu activation function. A dropout and fnally another dense layer is added to classify the normal kidney as well as cysts, tumors, and stones.
The total number of parameters in our modifed VGG16 is 17,482,492 out of which 5,127,612 are the trainable parameters and 12,354,880 are the non-trainable parameters. Table 3.10 shows the number of parameters of the diferent models used in our study.

VGG16 continues the characteristics of the classical network's simple structure, expands the network's depth through the fexible use of 3 × 3 convolution and successfully improves network performance.[28]

Table 2.2: Number of parameters of diferent models.

Model	Total Parameter	Trainable parameter
VGG16	17,482,492	5,127,612
Resnet50	24,767,684	2,234,692
Inception v3	22,327,396	1,575,236
CCT	407,365	407,365
EANet	355,140	355,140
SWIN	151,668	150,612

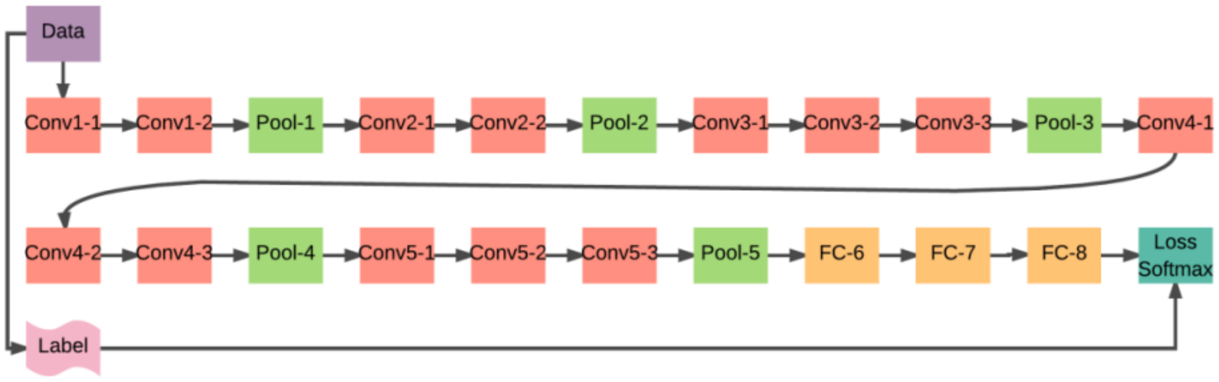


Figure 2.3: The network structures of VGG16 [4].

2.5 Classification using ResNet-50

ResNet-50 is a specific variant of ResNet, which is a convolutional neural network known for its depth. ResNet-50 specifically consists of 48 convolutional layers, along with 1 MaxPool layer and 1 Average Pool layer. The detailed architecture of ResNet-50 can be observed in Figure 2.4.

ResNet is based on the deep residual learning framework. It solves the problem of the vanishing gradient problem even with extremely deep neural networks. Resnet-50, despite having 50 layers has over 23 million trainable parameters which is very much smaller than existing architectures. The reasoning behind its performance is still open to discussions, but the simplest way to understand is to explain residual blocks and how these blocks work.

Let us consider a neural network block, whose input is x, where we would like to learn the true distribution H(x).
Let us denote the difference (or residual) between this as:

$$R(x) = Output - Input = H(x) - x \tag{2.1}$$

Rearranging it we get,

$$H(x) = R(x) + x \tag{2.2}$$

The residual block is trying to learn the true output, H(x). Taking a closer at the image above, we realize that since we have an identity connection coming due to x, the layers are learning the residual, R(x). The layers in a traditional network are learning true output (H(x)) while the layers in a residual network are learning the residual (R(x)). Also, it is observed that it is easier to learn the residual of the output and input rather than the input only. In this manner, the identity residual model allows for the reuse of activation functions from previous layers since they are skipped and add no complexities to the architecture. The ResNet-50 network model comprises 50 convolutional layers, three fully connected layers and five pooling layers.
In our experiment, the 50-layer ResNet-50 model was tweaked in the last few layers by using the last layer of the original ResNet-50 model, and we added average pooling, fattening, and a dense layer with a relu activation function. A dropout and fnally another dense layer is added to classify the normal kidney as well as cysts, tumors, and stones. The total number of parameters in our modifed ResNet-50 is 24,767,684 out of which 2,234,692 are the trainable parameters and 22,532,992 are the non-trainable parameters.[5]

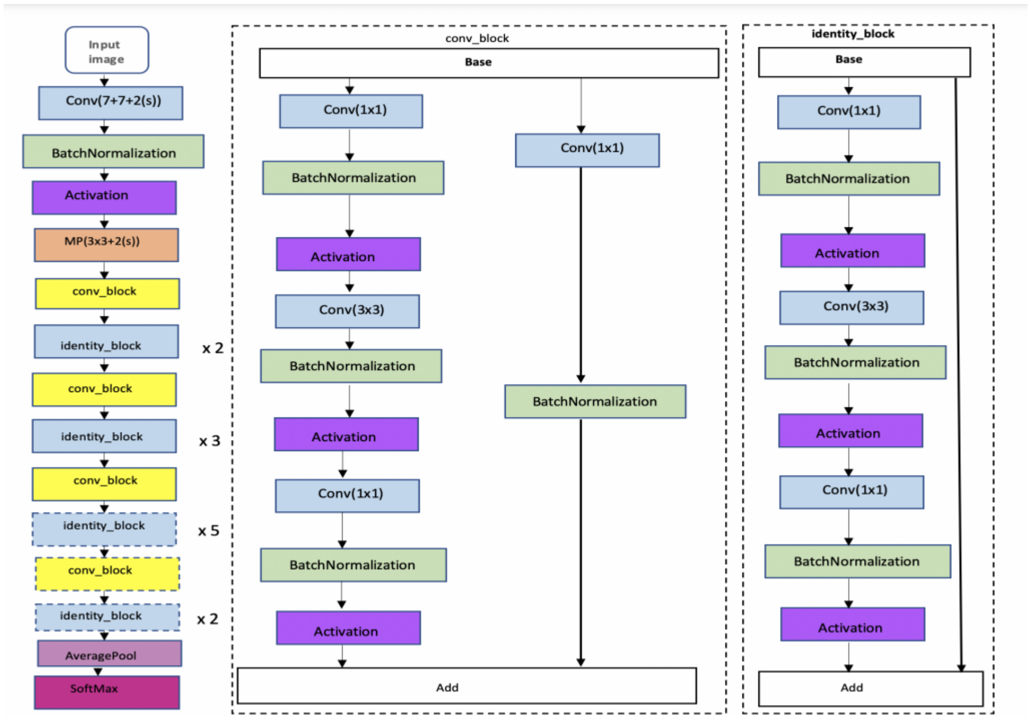


Figure 2.4: ResNet-50 architecure [5].

2.6 Classification using Inception-V3

A variant of the Inception family neural network, Inception v3 based on Depthwise Separable Convolutions, is used in our study to classify images. [9] The Inception v3 model, as described in our study, follows a specific structure. It executes the first seven layers and the AuxLogits and Logits area sequentially attached Fig. 2.3. The AuxLogits area is used during training of the network to reduce the effect of the vanishing gradient. Average pooling layers have a similar function as max-pooling layers, and are used downsampling to reduce overfi ing. The con- catenation layer chains output tensors of the same height and width. The dropout layer randomly disables the weights of a fully connected layer. The other areas are processed in parallel [6].

In our experiment, we made modifications to the original Inception v3 model. Specifically, we unfroze the last two layers to allow for further training. We added additional layers, including average pooling, flattening, a dense layer, and a dropout layer, followed by another dense layer for the classification task. The Inception v3 model has a total of 21,802,784 parameters, out of which 1,050,624 parameters are trainable. The remaining 20,752,160 parameters are non-trainable.

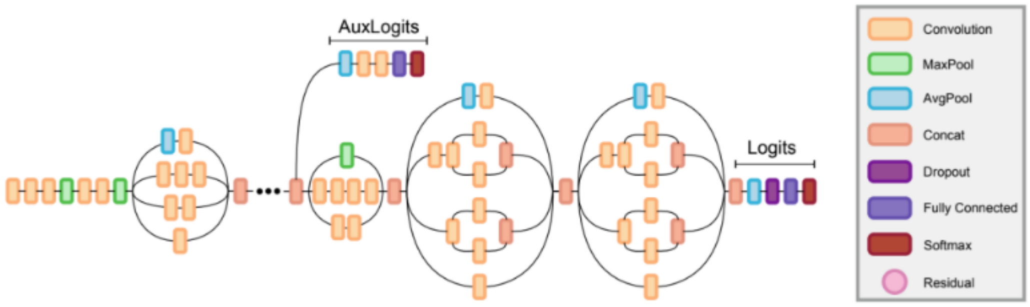


Figure 2.5: The CNN architecture of the Inception V3 model [6].

2.7 Vision Transformers

The rise of Vision Transformers, commonly referred to as ViTs, has generated substantial attention and demonstrated remarkable capabilities in the domain of computer vision. While Convolutional Neural Networks (CNNs) have traditionally been the dominant approach for image processing tasks, Vision Transformers introduce an alternative architecture that capitalizes on self-attention mechanisms, offering a promising direction for exploration and progress in the field. ViT, short for Vision Transformer, employs a unique approach by treating images as sequences of tokens. It divides an input image into patches, which are then transformed into vectors known as patch embeddings or patch tokens using linear projection(note that the sequence length remains the same throughout the network). These patch embeddings are subsequently processed through multiple Transformer blocks. The overall architecture of ViT, is depicted in Figure 2.6. The key components of ViT include tokenization, position embedding, multi-head self-attention (MHSA), feed-forward network (FFN), and layer normalization (LN). Notably, MHSA and FFN work together to form a transformer block, and we provide further details on these components below[33].

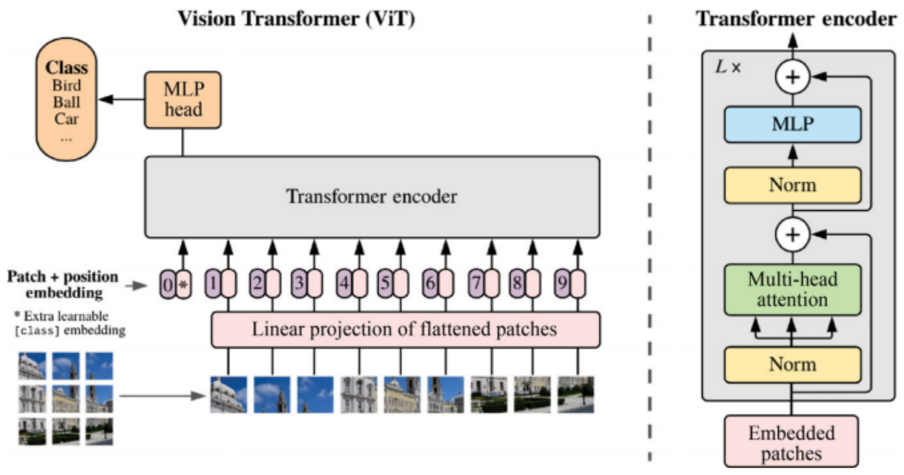


Figure 2.6: Framework of ViT (left) and typical pipeline of a transformer encoder (right) [7].

1 Tokenization

In the context of analyzing an input image $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ and its corresponding labels $\mathbf{Y}$, a crucial step involves reshaping $\mathbf{X}$ into a sequence of flattened 2D image patches denoted as $\mathbf{X}_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$, where N represents the number of patches and P signifies the patch size. this smaller patches, are then flattened to create a sequence of 2D patch embeddings. Each patch embedding (flattened patch) is then passed through a linear layer, the linear projection involves multiplying the flattened patch embeddings by a learnable weight matrix resulting in a d-dimensional vector. This matrix is typically represented by a linear layer, which performs a linear transformation on the input. The purpose of this linear projection is to map the flattened patch embeddings to a lower-dimensional feature space, where more meaningful and discriminative representations can be learned.

To facilitate the subsequent classification task, a special token called the *class* token is introduced at the beginning of the token sequence. The *class* token is generated from the outputs of the final attention block and is fed into the classification head, which predicts the class of the input. Notably, the initial state of the *class* token, which is inserted prior to token processing, is a parameter that can be learned by the model.

The purpose of the *class* token is to attend to the most relevant features or regions within the image, enabling effective classification[33, 34].

2 Position Embedding

Additionaly to the class token, several position tokens (embeddings) in the form of predefined or learnable vectors are added to patch embeddings which have the same dimension d model as the embeddings, so that the two can be summed, to record extra meaningful information and preserve relative positions of image patches for inference. Together, the input is formulated as follows [34, 33]:

$$z_0 = [x_{cls}; x_p^1 \cdot \mathbf{E};\ ...\ ; x_p^N \cdot \mathbf{E}] + [\mathbf{E}_{pos}^{cls}; \mathbf{E}_{pos}^1;\ ...\ ; \mathbf{E}_{pos}^N] \tag{2.3}$$

where $x_{cls} \in \mathbb{R}^D$ is the class token, $\mathrm{E} \in \mathbb{R}^{N \times (P^2 \cdot C) \times D}$ is a linear projection of each patch $\mathbf{X}_p$, and $\mathbf{E}_{pos}^i \in \mathbb{R}^D$ is the learnable position embedding for the i-th token.

Then, joint patch and position embeddings are processed with transformer blocks [33].

sine and cosine functions with different frequencies are employed in this work to encode the positional information.

$$PE_{(pos,2_i)} = \sin(pos/10000^{2i/d_{model}}) \tag{2.4}$$

$$PE_{(pos,2_i+1)} = \cos(pos/10000^{2i/d_{model}}) \tag{2.5}$$

where pos represents the position of the token within the sequence and i represents the index of the dimension in the embedding vector. That is, each dimension of the positional encoding corresponds to a sinusoid. The wavelengths form a geometric progression from 2π to $10000 \cdot 2\pi$. We chose this function because we hypothesized it would allow the model to easily learn to attend by relative positions, since for any fixed offset k, PE_{pos+k} can be represented as a linear function of PE_{pos}. [35].

3 Attention

The attention mechanism in Vision Transformers involves an attention function that takes a query vector and a set of key-value vector pairs as inputs, and produces an output vector. This attention function, known as Scaled Dot-Product Attention, was introduced by Vaswani et al. (2017). It calculates the relevance or similarity between the query and key vectors by taking their dot product. the equation can be formally expressed as:

$$Attention(\mathbf{Q},\ \mathbf{K},\ \mathbf{V}) = Softmax\left(\frac{\mathbf{Q}\cdot\mathbf{K}^T}{\sqrt{d_k}}\right) \tag{2.6}$$

where **Q**, **K**, and **V** represent respectively the set of queries, keys, and values. The initial patch representations are transformed into query, key, and value vectors grouped in matrices, these transformations are typically achieved using linear projections. d_Q, d_k represents the dimension of the queries and the keys, while the values have dimension d_v. The dot product between the query vectors and the key vectors measures the similarity between the patch associated with the query and the patch associated with the key.

for each patch, its query vector is dot-producted with the key vectors of all other patches to obtain a set of similarity scores. Higher dot product values indicate a higher similarity, while lower values indicate a lower similarity. the scores obtained from the dot product are scaled by $\frac{1}{\sqrt{d_k}}$ to compensate for the fact that the dot products can reach large values in magnitude when d_k is large. Large values would saturate softmax and make its gradients very small. The scaled scores are then passed through a softmax function to obtain attention weights, which represent the importance of each patch with respect to the others. The attention weights derived from the similarity scores are used to compute a weighted sum of the value vectors, which produces the final output of the self-attention mechanism. The main advantage of attention, when compared to convolutions, is the ability to capture long distance dependencies between tokens, which is something convolutions fail to do because of the local nature of the convolution operator. [36].

4 Multi-Head Attention

Vaswani et al. (2017) introduced a key step in the Transformer model where the input (z_0) undergo a linear projection using learnable weight matrices. This projection step transforms the input vectors into new representations **Q**, **K** and **V** with dimensions d_Q, d_K, and d_V, respectively. Specifically, the input embeddings or features are multiplied by specific weight matrices $\mathbf{W}^Q$, $\mathbf{W}^K$, $\mathbf{W}^V$, which are randomly initialized at the beginning of the training process.

Importantly, this linear projection is performed not just once, but h times. Each attention head in the model has its own set of learnable weight matrices for the linear projection. By conducting the projection h times, the resulting transformed representations from each attention head are simultaneously passed to the Attention function. This collective process is known as MultiHead Attention, where multiple attention heads work in parallel to capture different aspects of the input information [36].

After the parallel processing, the resulting matrices are concatenated and linearly projected, using, again, a learnable matrix. Parallel processing enables the model to efficiently process information from different representations of the inputs. Formally

$$MultiHead(\mathbf{Q},\mathbf{K},\mathbf{V}) = Concat(head_1,\ \dots\ ,head_h)\mathbf{W}^O, \tag{2.7}$$

$$Where\ head_i = Attention(\mathbf{Q}\mathbf{W}_i^Q,\mathbf{K}\mathbf{W}_i^K,\mathbf{V}\mathbf{W}_i^V),$$

where $\mathbf{W}_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $\mathbf{W}_i^V \in \mathbb{R}^{d_{model} \times d_v}$, and $\mathbf{W}^O \in \mathbb{R}^{hd_v \times d_{model}}$ are the learnable matrices used to linearly project the inputs and the result of the Attention operation. Finally, we call Self-Attention the special case where $\mathbf{K} = \mathbf{V}$, and –analogously– Multi-Head Self-Attention (MSA) a Multi-Head Attention in the case where $\mathbf{K} = \mathbf{V}$.

The output of the attention function for the embedding is the sum of N value vectors multiplied with their corresponding normalized weights. The process is conducted for all embeddings and the attention function results in an output as follows:

$$Attn(\mathbf{X}, W^K, W^Q, W^V) = softmax\left(\frac{\mathbf{Q} \cdot \mathbf{K}^T}{\sqrt{d_k}}\right) \times \mathbf{V}, \tag{2.8}$$

where $K = XW^K, Q = XW^Q, V = XW^V$

One MHSA contains h parallel attention functions or attention heads. The h different outputs are concatenated and projected with a linear transformation to produce the output of MHSA [33].

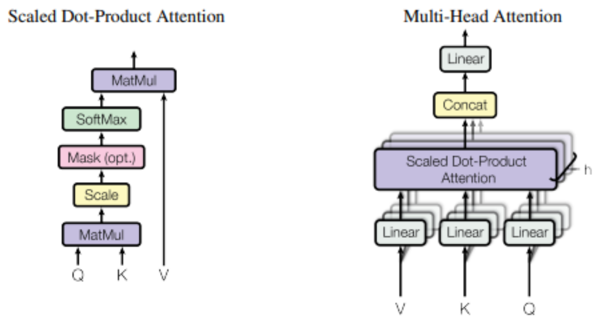


Figure 2.7: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel [7].

5 Layer Normalization

Layer normalization normalizes each embedding x in the sequence $\mathbf{X}$ separately as follows [33]:

$$LN(x) = \frac{x - \mu}{\sigma} \circ \gamma + \beta \tag{2.9}$$

It is applied both after the MSA and the MLP residual connections[36].

6 The MLP block and the overall Transformer block

The result of computing self-attention for the inputs is then passed to a Multi-Layer Perceptron (MLP) block with one hidden layer. it is responsible for processing and transforming the patch representations, allowing the model to capture complex relationships and higher-level features.

The MLP weights are learned during the training process of the Vision Transformer. The model optimizes the weights using gradient-based optimization algorithms, such as stochastic gradient descent (SGD) or Adam, to minimize a specific loss function.

Formally, the MLP block can be stated as follows: given an input x, learnable weights and biases W1, b1, W2, b2, and an non-linear activation function ρ, the MLP block can be expressed as [36]:

$$MLP(x) = \rho(x\mathbf{W1} + \mathbf{b1})\mathbf{W2} + \mathbf{b2} \tag{2.10}$$

In summary, the Transformer block consists of an input xl to the l-th block, a multi-head self-attention block (MSA), an MLP block, and a layer normalization block (LN). Combining these components together forms the complete structure of the Transformer block.

$$\begin{aligned} x_l' &= LN(x_l + MSA(x_l)) \\ x_{l+1} &= LN(x_l' + MSA(x_l')) \end{aligned} \tag{2.11}$$

2.8 Transformer Varients

1 Self-Attention and External Attention

The vision transformer architecture uses self-attention to sequences of image patches. The sequence of image patches is the input to the multiple transformer block in this case, which uses the multihead attention layer as a self-attention mechanism. One variant of the Vision Transformer EANet is shown in Figure 2.8. EANet utilizes external attention, based on two external, small, learnable, and shared memories, Mk and Mv. The purpose of EANet is to drop patches that contain redundant and useless information and hence improve performance and computational efficiency. External attention is implemented using two cascaded linear layers and two normalization layers. EANet computes attention between input pixels and external memory unit [9].

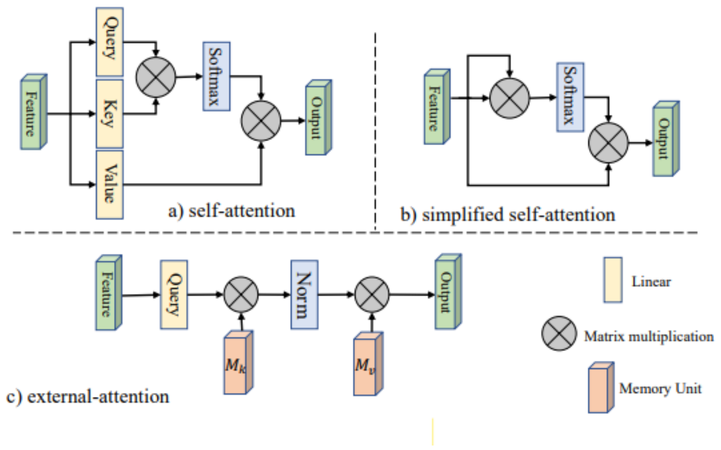


Figure 2.8: Self-attention versus external-attention [8].

We first revisit the self-attention mechanism (see Figure 2.8.a)). Given an input feature map $F \in \mathbb{R}^{N \times d}$, where N is the number of elements (or pixels in images) and d is the number of feature dimensions, self-attention linearly projects the input to a query matrix $Q \in \mathbb{R}^{N \times d'}$, a key matrix $K \in \mathbb{R}^{N \times d'}$, and a value matrix $V \in \mathbb{R}^{N \times d}$. Then self-attention can be formulated as:

$$A = (\alpha)_{ij} = softmax(QK^T), \tag{2.12}$$

$$F_{out} = AV, \tag{2.13}$$

where $A \in \mathbb{R}^{N \times N}$ is the attention matrix and $\alpha_{i,j}$ is the pair-wise affinity between (similarity of) the i-th and j-th elements.

A common simplified variation (Figure 2.8b) of selfattention directly calculates an attention map from the input feature F using:

$$A = softmax(FF^T), \tag{2.14}$$

$$F_{out} = AF. \tag{2.15}$$

Here, the attention map is obtained by computing pixel-wise similarity in the feature space, and the output is the refined feature representation of the input. However, even when simplified, the high computational complexity of $O(dN^2)$ presents a significant drawback to use of self-attention.

Self-attention can be viewed as using a linear combination of self values to refine the input feature. However, it is far from obvious that we really need $N \times N$ self attention matrix and an N element self value matrix in this linear combination. Furthermore, self-attention only considers the relation between elements within a data sample and ignores potential relationships between elements in different samples, potentially limiting the ability and flexibility of self-attention.

Thus, we propose a novel attention module named external attention, which computes attention between the input pixels and an external memory unit $M \in \mathbb{R}^{S \times d}$, via:

$$A = (\alpha)_{i,j} = Norm(FM^T), \tag{2.16}$$

$$F_{out} = AM. \tag{2.17}$$

Unlike self-attention, $\alpha_{i,j}$ in Equation (2.16) is the similarity between the i-th pixel and the j-th row of M, where M is a learnable parameter independent of the input, which acts as a memory of the whole training dataset. A is the attention map inferred from this learned dataset-level prior knowledge; it is normalized in a similar way to self-attention (see Section 3.2). Finally, we update the input features from M by the similarities in A.

In practice, we use two different memory units M_k and M_v as the key and value, to increase the capability of the network. This slightly changes the computation of external attention to

$$A = Norm(FM_k^T), \tag{2.18}$$

$$F_{out} = AM_v. \tag{2.19}$$

The computational complexity of external attention is $O(dSN)$; as d and S are hyper-parameters, the proposed algorithm is linear in the number of pixels. In fact, we find that a small S. Thus, external attention is much more efficient than selfattention, allowing

its direct application to large-scale inputs. We also note that the computation load of external attention is roughly equivalent to a 1 × 1 convolution.
In this study, we used 256 patches per image, with an embedding dimension of 64, MLP dimension of 64, and 8 transformer blocks. The batch size was set to 128, and the number of training epochs was 50. num-heads = 4, s = 16

2 Compact Convolutional Transformers

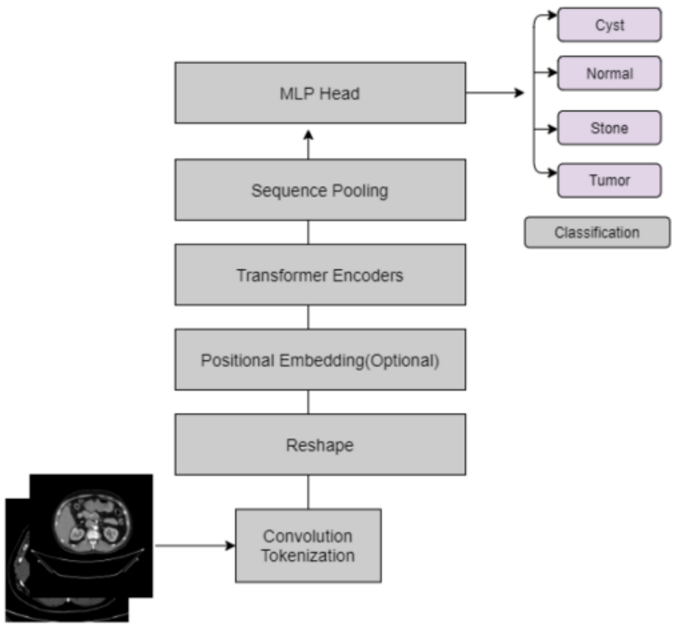


Figure 2.9: the Compact Convolutional Transformer (CCT) architecture [9].

The CCT is the most recent version of the Vision Transformer (Vit), which is a compact model for image processing problems[37].

While utilizing a standard transformer encoder, the CCT introduces two distinctive features: a convolutional image encoding block and a sequence pooling layer, which replace certain operations found in traditional transformer approaches.

Figure 2.9 provides an overview of the Compact Convolutional Transformer (CCT) approach.

To begin with, the CCT employs a convolutional block, consisting of multiple convolutional layers, to embed the input image into a latent space. Unlike the conventional practice of dividing images into non-overlapping patches, our approach utilizes the convolutional block to preserve information at patch boundaries and capture aggregate information from all image regions. This choice not only results in more informative inputs for the transformer encoder but also introduces inductive biases to enhance training efficiency, particularly on smaller-sized datasets.

Following the convolutional block, the feature maps are row concatenated into a vector, which serves as the input tokens for the transformer encoder. To enable the transformer to understand the spatial relationship between tokens, positional embedding is applied to each token.

The transformer encoder consists of two transformer encoder layers, each equipped with a multi-headed attention mechanism that captures long-range dependencies within the input. These transformer encoder layers follow the typical design of attention-based layers.

After the transformer encoder, sequence pooling is performed on the outputs. This pooling operation smooths the sequence of outputs and enables the MLP Head to accurately classify the classes of interest. Notably, sequence pooling eliminates the need for an additional classification token, which is commonly used in other transformer models. As a result, the model no longer needs to track the classification token throughout its layers, simplifying the architecture. [38].

For an image (y) the convolutional tokenization procedures will be:

$$y_0 = MaxPool(ReLU(Conv2D(y))) \tag{2.20}$$

The Conv2D feature maps generated by the CCT undergo scaling using the maxpool layer and are activated by the ReLU activation function.

The inclusion of the sequence pooling layer in the CCT enables the network to evaluate the sequential embedding of latent spaces produced by the transformer encoder. This pooling operation captures meaningful information from different regions of the input images. It is a mapping transformation technique employed to pool the entire sequence of data and extract valuable insights; it is denoted as $T : R(i \times n \times j) \longrightarrow R(i \times j)$. [39].

This procedure can be described as:

$$y_L = f(y_0) \in \mathbb{R}^{(i \times n \times j)} \tag{2.21}$$

where the transformer encoder of a layer is denoted as L and its output is denoted as y_L or $f(y_0)$. Furthermore, a mini-batch size denoted by i, j is taken as the embedding dimension, and n indicates the sequence length. Then, y_L is fed to a linear layer $g(y_L) \in \mathbb{R}^{(i \times 1)}$ and the Softmax activation function 2.22 is utilized.

$$y_L^{'} = softmax(g(y_L)^T) \in \mathbb{R}^{(i \times 1 \times n)} \tag{2.22}$$

The output can be calculated as:

$$output; o = y_L^{'} y_L = softmax(g(y_L)^T \times x_L \in \mathbb{R}^{(i \times 1 \times j)} \tag{2.23}$$

After pooling of the second dimension, output $(o) \in \mathbb{R}^{(i \times 1 \times j)}$ is attained as an output. After passing through a linear classification layer, the images are categorized [39].

In the case of the Compact Convolutional Transformer (CCT), the input images were reshaped to a size of 32. We employed two convolutional layers in the model, and the embedding dimension was set to 128. During training, a batch size of 128 was used, and the training process was performed over 20 epochs.

3 Swin Transformers

The Swin transformer is a hierarchical transformer that uses shifting-window MSA and includes four stages as shown in Figure 2.10 [40].
It first splits an input image into non-overlapping patches by a patch splitting module, like ViT. Each patch is treated as a "token" and its feature is set as a concatenation of the raw pixel values. A linear embedding layer is applied on this raw-valued feature to project it to an arbitrary dimension (denoted as C). Several Transformer blocks with modified self-attention computation (Swin Transformer blocks) are applied on these patch tokens. The Transformer blocks maintain the number of tokens $(\frac{H}{4} \times \frac{W}{4})$, and together with the linear embedding are referred to as "Stage 1".

To produce a hierarchical representation, the number of tokens is reduced by patch merging layers as the network gets deeper. The first patch merging layer concatenates the features of each group of 2×2 neighboring patches, and applies a linear layer on the 4C-dimensional concatenated features. This reduces the number of tokens by a multiple of $2 \times 2 = 4$ $(2 \times downsampling of resolution)$, and the output dimension is set to $2C$. Swin Transformer blocks are applied afterwards for feature transformation, with the resolution kept at $\frac{H}{8} \times \frac{W}{8}$. This first block of patch merging and feature transformation is denoted as "Stage 2". The procedure is repeated twice, as "Stage 3" and "Stage 4", with output resolutions of $\frac{H}{16} \times \frac{W}{16}$ and $\frac{H}{32} \times \frac{W}{32}$, respectively. These stages jointly produce a hierarchical representation,with the same feature map resolutions as those of typical convolutional networks, e.g., VGG and ResNet. As a result, the proposed architecture can conveniently replace the backbone networks in existing methods for various vision tasks [10].

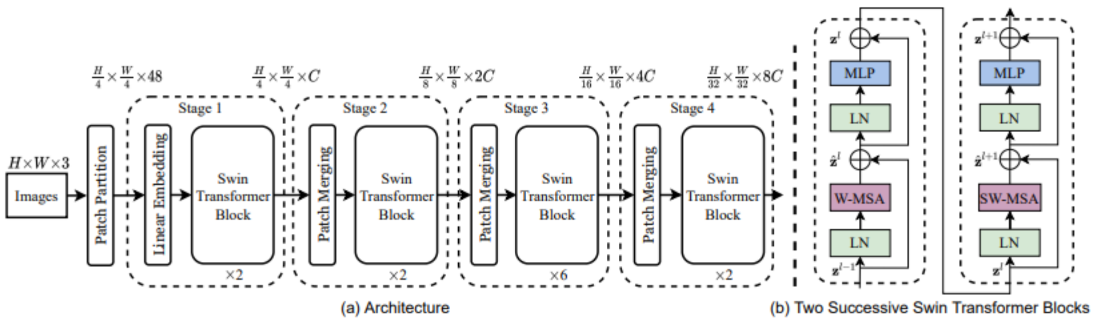


Figure 2.10: The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks (notation presented with Eq. (3)). W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively[10].

Swin Transformer block

The Swin Transformer architecture introduces a modification to the standard Transformer block by replacing the multi-head self-attention (MSA) module with a module based on shifted windows. This new module, which will be further explained in the next section, is combined with other layers that remain unchanged. In Figure 2.10(b), the structure of a Swin Transformer block is depicted. It consists of a shifted window-based MSA module, followed by a 2-layer MLP with GELU nonlinearity in between. Layer normalization (LN) is applied before each MSA module and each MLP, ensuring stable training. Additionally, a residual connection is applied after each module, allowing the flow of information through the block [10].

4 Shifted Window based Self-Attention

Both the original Transformer architecture proposed by Vaswani [35] and its adaptation for image classification introduced by [7] rely on global self-attention. Global self-attention involves computing the relationships between a token and all other tokens in a sequence. However, this global computation poses a challenge due to its quadratic complexity, meaning that the computational requirements increase rapidly with the number of tokens. As a result, this approach becomes impractical for vision problems that involve a large number of tokens, such as dense prediction tasks or the representation of high-resolution images.

Self-attention in non-overlapped windows

To achieve more efficient modeling, we introduce the concept of computing self-attention within local windows. These windows are arranged in a non-overlapping manner to evenly partition the image. Each window encompasses a set of patches with dimensions of $M \times M$. By employing this window-based approach, we can significantly reduce the computational complexity compared to the global multi-head self-attention (MSA) module.

In terms of computational complexity, let's consider an image composed of $h \times w$ patches. The global MSA module involves computing self-attention between all pairs of patches, resulting in a quadratic complexity. However, with the window-based approach, the computational complexity is reduced due to the limited scope of attention within each window.

$$\Omega(\mathbf{MSA}) = 4hwC^2 + 2(hw)^2C, \tag{2.24}$$

$$\Omega(\mathbf{W\text{-}MSA}) = 4hwC^2 + 2M^2hwC, \tag{2.25}$$

where the former is quadratic to patch number hw, and the latter is linear when M is fixed (set to 7 by default). Global self-attention computation is generally unaffordable for a large hw, while the window based self-attention is scalable.

Shifted window partitioning in successive blocks

The window-based self-attention module lacks connections across windows, which limits its modeling power. To introduce cross-window connections while maintaining the efficient computation of non-overlapping windows, we propose a shifted window partitioning approach which alternates between two partitioning configurations in consecutive Swin Transformer blocks.

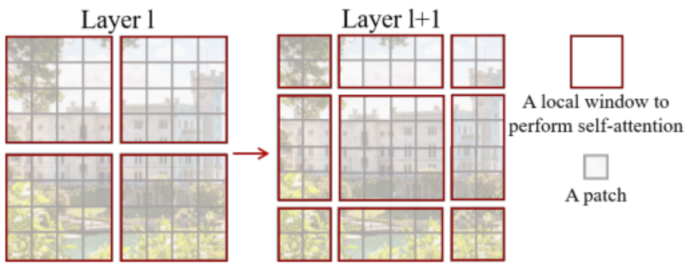


Figure 2.11: An illustration of the shifted window approach for computing self-attention in the proposed Swin Transformer architecture. In layer l (left), a regular window partitioning scheme is adopted, and self-attention is computed within each window. In the next layer l + 1 (right), the window partitioning is shifted, resulting in new windows. The self-attention computation in the new windows crosses the boundaries of the previous windows in layer l, providing connections among them [10].

As illustrated in Figure 2.11, the first module uses a regular window partitioning strategy which starts from the top-left pixel, and the 8×8 feature map is evenly partitioned into 2×2 windows of size 4×4 ($M = 4$). Then, the next module adopts a windowing configuration that is shifted from that of the preceding layer, by displacing the windows by $([\frac{M}{2}], [\frac{M}{2}])$ pixels from the regularly partitioned windows. With the shifted window partitioning approach, consecutive Swin Transformer blocks are computed as

$$\hat{z}^l = \textbf{W-MSA}(LN(z^{l-1}) + z^{l-1}, \tag{2.26}$$

$$z^l = \textbf{MLP}(\textbf{LN}(\hat{z}^l)) + \hat{z}^l, \tag{2.27}$$

$$\hat{z}^{l+1} = \textbf{SW-MSA}(\textbf{LN}(z^l)) + z^l, \tag{2.28}$$

$$z^{l+1} = \textbf{MLP}(\textbf{LN}(\hat{z}^{l+1})) + \hat{z}^{l+1} \tag{2.29}$$

where $\hat{z}^l$ and z^l denote the output features of the **(S)W-MSA** module and the MLP module for block l, respectively.

In our experiment, we utilized an image patch size of 2. The number of attention heads was set to 8, and the embedding dimension was 64. We employed 256 MLP layers, and the attention window size was 2. The image dimension was fixed at 32. For training, a batch size of 128 was used, and the model was trained for 100 epochs.

2.9 Handcrafted features

1 Local binary pattern operator (LBP)

Local Binary Patterns (LBP) is a technique commonly used in image processing to extract meaningful features. It involves applying the LBP operator to each pixel, which generates a binary value denoted as $S(f_p - f_c)$. This value is determined by comparing the pixel with its center pixel (f_c) and its surrounding pixels f_p within a 3×3 window. An illustrative example of this operation can be seen in Figure 2.12. To obtain the LBP values, the differences between the neighboring pixels of each pixel are binarized using a step function specified in Eq 2.30.

$$LBP_{P,R}(f_c) = \sum_{P=0}^{P-1} \mu(f_p - f_c)2^P, \mu(y) = \begin{cases} 1, & y \geq 0 \\ 0, & y < 0 \end{cases} \tag{2.30}$$

In the equation for $LBP_{P,R}$, R specifies the radius and specifies the distance of neighboring pixels from the center pixel, while P denotes the number of neighbor pixels included in the process. As in the example below Figure 2.12, radius R is taken as one 1, while neighbor number P is taken as eight 8 [11].

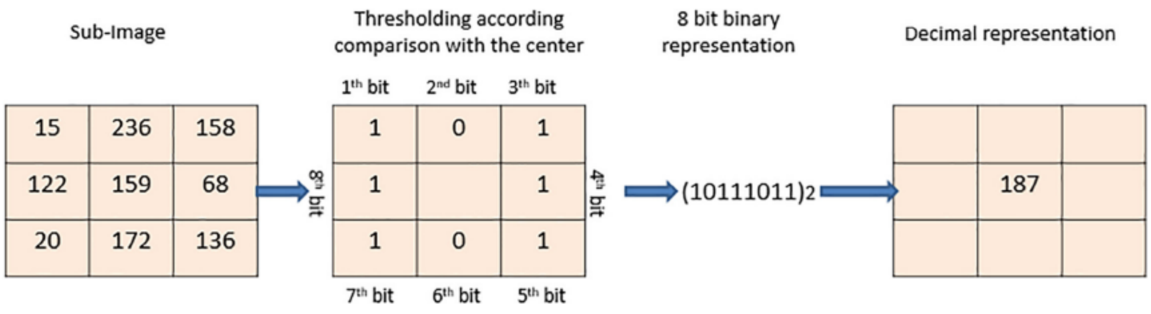


Figure 2.12: Calculating the original LBP code [11].

In our study, we employed Local Binary Patterns (LBP) as handcrafted features to extract relevant information. These features were subsequently fed into a k-Nearest Neighbors (kNN) classifier with K=3 for classification purposes.we used the same approach with HOG and BSIF handcrafted features

2 Histogram of Oriented Gradients (HOG)

In this section, we provide an overview of the HOG feature extraction method. HOG involves two primary computation units: the cell and the block. Figure 2.12 illustrates the relationship between these units.

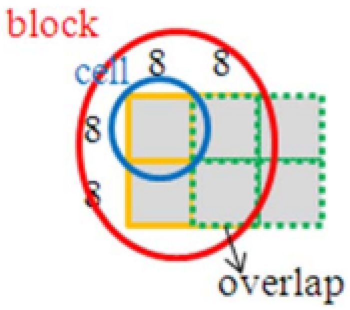


Figure 2.13: Cells and blocks used for HOG feature extraction [12].

Gradient Computation

To calculate the magnitude $m(x, y)$ and direction $\theta(x, y)$ for each pixel at coordinate (x, y), the following steps are performed. Let $f(x, y)$ represent the luminance value of the pixel located at coordinate (x, y). $f_x(x, y)$ and $f_y(x, y)$ are computed by the following equations:

$$f_x(x, y) = f(x + 1, y) - f(x - 1, y) \tag{2.31}$$

$$f_y(x, y) = f(x, y + 1) - f(x, y - 1). \tag{2.32}$$

Then magnitude $m(x, y)$ and direction $\theta(x, y)$ of the computed gradients are computed by

$$m(x, y) = \sqrt{f_x(x, y)^2 + f_y(x, y)^2}. \tag{2.33}$$

and

$$\theta(x, y) = arctan\frac{f_y(x, y)}{f_x(x, y)}. \tag{2.34}$$

Gradient Vote

Once the magnitude $m(x, y)$ and direction $\theta(x, y)$ are obtained, each pixel within the cell contributes to the computation of an orientation histogram based on the gradient orientation of the element centered on it. The orientations are evenly distributed from 0°

to 180° and divided into nine bins. The weight of each pixel, represented by α, can be calculated as follows:

$$\alpha = (n + 0.5) - \frac{b * \theta(x, y)}{\pi} \tag{2.35}$$

In this equation, n represents the bin to which the gradient direction $\theta(x, y)$ belongs, and b is equal to 9, indicating the total number of bins. To mitigate aliasing effects, the values of both the current bin and its neighboring bin are incremented. The incremented values m_n and $m_{nearest}$ can be given by

$$m_n = (1 - \alpha) * m(x, y) \tag{2.36}$$

$$m_{nearest} = \alpha * m(x, y) \tag{2.37}$$

respectively. Votes ($m0$ to $m8$) are accumulated into orientation bins over local spatial regions.

Normalization Computation

Ultimately, a histogram normalization computation is performed by combining all the histograms associated with a block, which is composed of four cells. The resulting normalized histogram can be achieved by applying the following expression:

$$v_i^n = \frac{v_i}{\sqrt{\|v\|_2^2 + \epsilon^2}} \tag{2.38}$$

where i is a number from 1 to 36 (four cells × nine bins), v_i is the vector corresponding to a combined histogram for a block region, $\|v\|_2^2 = v_1^2 + v_2^2 + ... + v_{36}^2$ and ϵ is a small constant to avoid dividing by zero. [29, 12].

3 Binary Statistical Image Features (BSIF)

The BSIF codes are represented as a histogram of pixel binary codes, it aims to capture texture information from images by analyzing local binary patterns. The size of the filter and the length of the bit strings are important factors in evaluating the BSIF .[30]

Binary Statistical Image Feature (BSIF) is a texture descriptor which is inspired by Local Binary pattern (LBP). BSIF uses a binary code to represent each pixel neighborhood in an image. The binarized feature is generated by convolving image with a set of linear filter. Thus, the response of a linear filter is binarized with a threshold at zero. To construct the linear filter, independent component analysis (ICA) is used. The statistical independence of the filter responses is maximizing by ICA from a training set of natural image patches. The BSIF extraction process is as follow: First, the response of the linear filter is constructed. Let X is image patch with size $l \times l$ pixels. W_i is a linear filter and S_i represents the response of the filter [31].

$$s_i = \sum W_i(u, v)X(u, v) = W_i^T x \tag{2.39}$$

Binarized feature b_i is obtained by:

$$b_i = \begin{cases} 1, \; if \; s_i > 0 \\ 0, \; otherwise \end{cases} \tag{2.40}$$

2.10 Lightweight neural networks

1 DLNet

Convolutional filters generation

Assuming that we are given an input image, denoted by I with the dimensions $M \times N$. We consider using discrete cosine transform (DCT) filter bank, which allows our network to have the property of data independency (i.e., in contrast to data-driven filters) [13].

$$f(u,v) = \delta(u)\delta(v)\sum_{i=1}^{M}\sum_{j=1}^{N} f(x,y)\cos[\frac{\pi(2i+1)u}{2M}]\cos[\frac{pi(2j+1)u}{2N}], \tag{2.41}$$

where

$$\delta(u) = \begin{cases} \frac{1}{\sqrt{M}}, & ifu = 0 \\ \sqrt{\frac{2}{M}}, & otherwise \end{cases} \tag{2.42}$$

$$\delta(v) = \begin{cases} \frac{1}{\sqrt{N}}, & ifv = 0 \\ \sqrt{\frac{2}{N}}, & otherwise \end{cases} \tag{2.43}$$

Convolution layer

The previously generated DCT filters are employed to carry out convolution operations. Let's assume that the 2D filter size is $k \times k$. The input image, denoted as I, is convolved with various 2D DCT bases using the following process

$$O^p = \{I * W_p\}_{p=1}^{L} \tag{2.44}$$

Where, $W_p \in \mathbb{R}^{k\times k}$, $p = 1, \ldots, L$ represents the set of 2D DCT bases or filters, and p_L stands for the number of filters. It is important to note that the resulting feature maps, denoted as O^p, have the same size as the input image I. This is achieved by zero-padding the borders of the image with a padding size of $(k-1)/2$ before performing the convolution [13].

Binary hashing

To prevent over-fitting in our network, we employ a binary hashing procedure on feature maps. This involves quantifying the filter responses after convolving the input image with DCT filters, resulting in feature maps with real-number values. To binarize each map, we set zero as the threshold, converting values higher than zero to one and leaving zero unchanged as shown in Eq 2.45 [13].

$$BIN(O^p) = \begin{cases} 1, & ifO^p > 0 \\ 0, & otherwise \end{cases} \tag{2.45}$$

The binarized feature maps are combined to form a single image denoted by D Figure. 2.13. In this case, every pixel in D will range from 0 to 2^{L1}. The Combination is done according to the next equation.

$$D = \sum_{p=1}^{L} 2^{p-1} BIN(O^p) \tag{2.46}$$

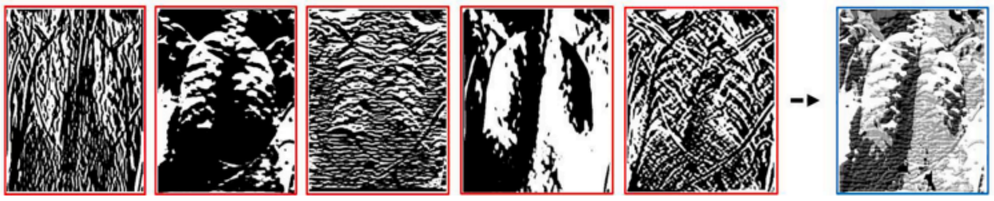


Figure 2.14: Binary hashing process, images with a red border are feature maps and the one with a blue border is the output of fusing those images [13].

Image encoding using local binary patterns

In the previous steps, our focus was on examining the responses of pixels to various filters. Now, in this step, our objective is to enhance the DLNet by incorporating local binary patterns (LBP) associated with different pixels in the image. To accomplish this, we compute binary codes for the pixels within the input image, as depicted in (Figure 2.11). Formally, for a pixel denoted as g_c, characterized by its coordinates $I(x_c, y_c)$ and situated within a neighborhood of p pixels, we generate the LBP code using the following equation [13].

$$LBP_{g_c} = \sum_{i=0}^{i=p-1} BIN(g_p - g_c)2^i \tag{2.47}$$

Block-wise filter response-based LBP histogram generation

The foundation of DLNet lies in jointly considering two kinds of crucial information: filter responses and local binary codes of image pixels. This combination allows us to achieve a representation that captures both aspects. To accomplish this, we extract a $2^M - dimensional$ histogram that combines the local binary patterns and filter responses. This histogram, known as $HISTL$, is and can be generated using Eq(2.46) [13]:

$$HISTL(w) = \sum_{a=1}^{u} LBP_{g_c}(Ind_w) \tag{2.48}$$

such that $u = |Indw|$ and w represents the index of the histogram. Ind_w is the set of pixels' spatial coordinates for which D is equal to w. Ind_w is defined by.

$$Ind_w \tag{2.49}$$

The operator is an assignment operator used to assign Indw with the spatial coordinates where $D = w$. To leverage the spatial relationship effectively, we adopt a block-wise approach to extract the histogram. This involves dividing both D and LBP_{g_c} into non-overlapping blocks. Subsequently, histograms are extracted from each block, and these individual histograms are concatenated into a single histogram. This concatenated histogram exhibits a certain degree of translation invariance due to its construction from

different blocks.
In our study, we employed DLNet as Lightweight neural networks to extract features. These features were subsequently fed into a k-Nearest Neighbors (kNN) classifier with K=3 for classification purposes.we used the same approach with MDFNet and DCTNet Lightweight neural networks [13].

2 MDFNet

MDFNet, which stands for Modified Deep Filtering Network, is a lightweight neural network that works similarly to DLNet, but with two key differences in its architecture.

The first differenceThe first differenc is used pca-data-driven filterst instead the DCT filter bank.

The second difference is in the choice of featureThe second difference is in the choice of feature map the lbp feature map is replaced by the gradient magnitude as feature map.

3 DCTNet

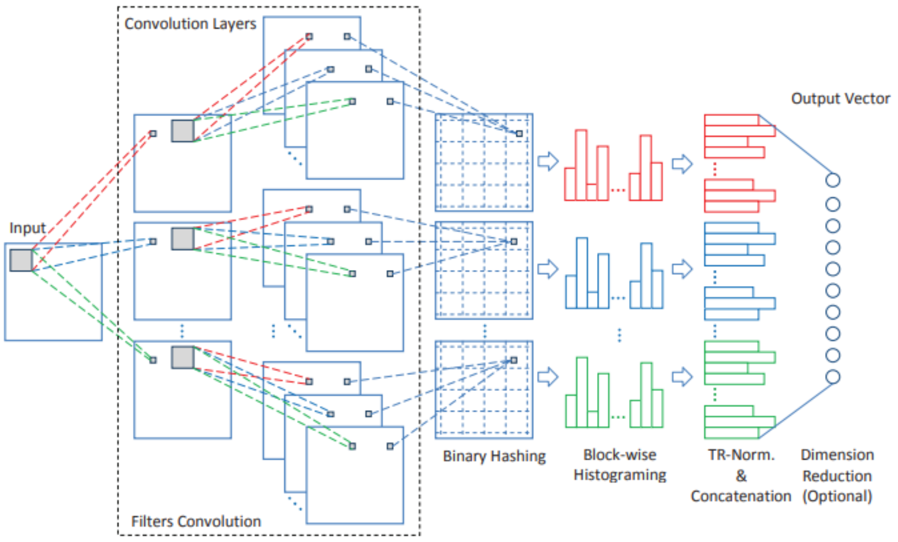


Figure 2.15: The block diagram of the proposed DCTNet [14].

DCTNet follows a structure similar to DLNet Figure 2.14 , with the first step being the Convolution Layer, and the third step involving Binarization and Block-wise Histograming. These steps are identical in both DLNet and DCTNet.Then, each of these D binarized "image" is partitioned into B non-overlapping blocks. Histogram of each block denotes by $H_b^d, b = 1, 2, \dots, B; d = 1, 2, \dots, D$ with bin $[0, 2^{PL-1}]$ is obtained as the input for histogram normalization layer that will be described in next section [14].

Histogram Tied Rank Normalization (TR Normalization)

The first stage of TR normalization uses tied rank principle that computes rank of a given vector x which produces a vector $\bar{x}$ that has a range from 1 to the length of x where each element $\bar{x}_i$ corresponds to the ascending order rank of x_i.In case of ties, their average rank is assigned to all ties which may produce non-integer values. Given **H** as the extracted block-wise histogram of a given face data, where $\mathbf{H} = \{H_b^d\}_{b=1,d=1}^{B,D}$. Each H_b^d is ranked with tied ranking without considering the bin with zero occurrence denoted by $\bar{H}_b^d$. This is because bin with zero occurrences is not a sample in histogram, it should be ignored in the ranking process. In order to make $\bar{H}_b^d$ to be more evenly distributed, we first apply square root on $\bar{H}_b^d$ forming $v_b^d = \sqrt{\bar{H}_b^d}$. Follow by $L2$ we obtain $\hat{v}_b^d$.The final TR normalized histogram feature vector is constructed by concatenating all $\hat{v}_b^d$ [14]

$$v = [\hat{v}_1^1, \hat{v}_2^1, \ldots, \hat{v}_B^1, \hat{v}_1^2, \ldots, \hat{v}_B^D] \in \mathbb{R}^{(2^P L)BD} \tag{2.50}$$

2.11 Conclusion

In conclusion, this chapter has presented an extensive exploration of multiple models and techniques employed in the analysis of kidney diseases within an academic framework. These models encompass convolutional neural networks (CNNs), transformers, lightweight neural networks, and handcrafted features. In the upcoming chapter, we will present the results obtained from applying these approaches to our datasets. These results will be carefully analyzed and interpreted, taking into account various evaluation metrics such as accuracy, loss, confusion matrix, and ROC curves. By examining these results, we will be able to assess the performance and efficacy of each model and technique in the context of kidney disease analysis.

Chapter 3

Experimental study

3.1 Introduction

3.2 Performance Evaluation Metrics

Although it is important in other fields, it is critical in the medical world to achieve high accuracy with an equal balance in the model's performance for each class. That is why accuracy should not be the only measure of performance. Other measures such as sensitivity, specificity, precision, recall, and $F1 - score$ should be calculated [22].

$$Accuracy = \frac{T_p + T_n}{T_p + F_p + F_n + T_n} \tag{3.1}$$

$$Sensitivity = \frac{T_p}{T_p + F_n} \tag{3.2}$$

$$Specificity = \frac{T_n}{T_n + F_p} \tag{3.3}$$

$$Precision = \frac{T_p}{T_p + F_p} \tag{3.4}$$

$$F1_{score} = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3.5}$$

Also to assess the effectiveness of the implemented models, we employed a range of performance evaluation metrics. These metrics encompassed training and loss curves, confusion matrices, as well as ROC curves. This comprehensive set of metrics allowed us to thoroughly evaluate the performance of the models.

1 Results

Our research aimed to conduct a comprehensive comparative study among various models, including CNN models, transformer variants, lightweight neural networks, and handcrafted features. The primary objective was to evaluate the performance of these models for the detection of kidney cyst, normal, stone, and tumor conditions. In order to expand the scope of our comparison and considering the limited availability of datasets specific to our target task, we introduced modifications to the existing dataset. These modifications included the addition of Gaussian noise, blur, occlusion, and variations in illumination (both increase and decrease). By examining the performance of each model under these modified conditions, we sought to gain insights into their robustness and effectiveness. Through our study, we aimed to identify the most suitable model or approach for accurate and reliable detection in the given medical context.

Experiment 1: performance using the original dataset

Based on the findings presented in Table 3.1, our comprehensive analysis revealed diverse performance outcomes across the different models when evaluated on our original dataset. The Inception-V3 model exhibited relatively lower performance, achieving an accuracy of 80.96%. Moderate performance was observed for EANet and Resnet-50, with accuracies of 81.35% and 89.13%, respectively. In contrast, CCT, VGG16, and Swin Transformers showcased higher accuracy rates of 96.73%, 90.38%, and 95.38%, respectively.

Interestingly, the handcrafted features, namely LBP, HOG, and BSIF, outperformed all other models with accuracies of 97.98%, 99.13%, and 99.51%, respectively. This superior performance can be attributed to the nature of the images in our dataset, which exhibited simplicity and were easier to process using traditional feature extraction techniques.

Moreover, it is important to highlight the impressive performance of transformer-based models, such as CCT and Swin Transformers. Given the inherent ability of transformers to effectively handle intricate data, we expect their performance to be notably superior to that of handcrafted features, particularly when applied to datasets comprising more challenging images.

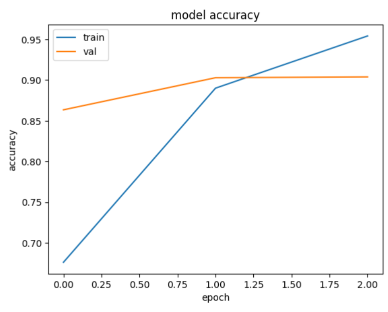
When analyzing Table 3.1, we can observe the performance of different models in detecting kidney diseases. The VGG-16 CNN model shows good recall values, particularly for kidney tumor classes, with a recall of 0.98. It also achieves recall values of 0.94, 0.78, and 0.97 for the cyst, normal, and stone classes, respectively. The CCT model performs well in detecting cyst and normal classes, with a recall of 0.98 for both, while achieving recall values of 0.97 and 0.94 for the stone and tumor classes, respectively. The MDFNet model demonstrates perfect recall (1) for cyst and normal classes, indicating its strong performance in detecting these types of images. Among the handcrafted features, HOG and BSIF achieve a recall of 1 specifically for the stone class. Recall values vary for different CNN models, with Resnet-50 achieving recall values of 0.88, 0.83, 0.93, and 0.95 for the cyst, normal, stone, and tumor classes, respectively. Inception-v3's recall values are 0.65, 0.86, 0.96, and 0.90 for the respective classes. EANet, SWIN, LBP, DCTNet, and DLNet also achieve varying recall values for the different classes. Higher recall values indicate a lower chance of misdiagnosing images belonging to the cyst, normal, stone, and tumor classes

When considering the different approaches in our analysis, we can observe notable variations in precision, F1 scores, and the Area Under the ROC Curve (AUC) for the different models. In the transfer-based approach, VGG16 shows higher precision compared to Inception V3 and Resnet50, achieving precision values of 0.98, 0.98, 0.82, and 0.83 for the respective classes. In the transformer-based approach, CCT stands out with superior precision values of 0.92, 0.97, 0.98, and 1 compared to EANet and SWIN. For the handcrafted features-based approach, BSIF, and HOG demonstrates better precision than LBP, with values of 0.99, 0.99, 0.98, and 1. Lastly, within the lightweight-based approach, MDFNet surpasses DCTNet and DLNet with precision values of 0.99, 0.98, 1, and 1 for the respective classes. In terms of F1 scores, VGG-16 achieves the highest scores among the cyst, normal, stone, and tumor classes, with values of 0.96, 0.87, 0.89, and 0.90, respectively. CCT and SWIN demonstrate strong F1 scores of 0.95, 0.98, 0.97, and 0.96, while HOG and BSIF exhibit impressive F1 scores of 0.99 for all classes. In the lightweight-based approach, MDFNet stands out with excellent F1 scores of 0.99 for all classes. Regarding the AUC, CCT, VGG16, and SWin Transformers show higher values compared to Resnet50, EANet, and Inception v3. The AUC values are closer to 1 for the Kidney Cyst, Normal, Stone, and Tumor categories in the BSIF and MDFNet models.

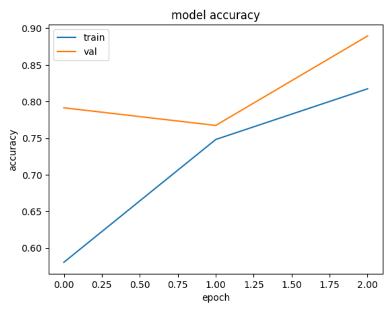
Table 3.1: The performance measures for all the models on the ORIGINAL dataset

Models	Accuracy	Class	Precision	Recall(Sensitivity)	F1 Score	AUC
VGG16	90.38%	Cyst	0.98	0.94	0.96	0.997
		Normal	0.98	0.78	0.87	0.993
		Stone	0.82	0.97	0.89	0.991
		Tumor	0.83	0.98	0.90	0.983
Resnet	89.13%	Cyst	0.90	0.88	0.89	0.989
		Normal	0.97	0.83	0.90	0.991
		Stone	0.84	0.93	0.88	0.992
		Tumor	0.85	0.95	0.90	0.987
Inception v3	80.96%	Cyst	0.98	0.65	0.78	0.984
		Normal	0.70	0.86	0.77	0.956
		Stone	0.80	0.96	0.87	0.984
		Tumor	0.76	0.90	0.82	0.975
CCT	96.73%	Cyst	0.92	0.98	0.95	0.986
		Normal	0.97	0.98	0.98	0.997
		Stone	0.98	0.97	0.97	0.992
		Tumor	1	0.94	0.96	0.995
EANet	81.35%	Cyst	0.98	0.74	0.85	0.985
		Normal	0.96	0.76	0.85	0.975
		Stone	0.84	0.95	0.89	0.987
		Tumor	0.47	0.88	0.61	0.906
SWIN	95.38%	Cyst	0.96	0.94	0.95	0.992
		Normal	0.98	0.98	0.98	0.998
		Stone	0.96	0.98	0.97	0.998
		Tumor	0.96	0.97	0.96	0.996
LBP	97.98%	Cyst	0.97	0.94	0.95	0.975
		Normal	0.94	0.94	0.94	0.961
		Stone	0.92	0.97	0.94	0.953
		Tumor	0.95	0.94	0.94	0.962
HOG	99.13	Cyst	0.98	0.99	0.99	0.989
		Normal	0.99	0.99	0.99	0.994
		Stone	0.99	1	0.99	0.995
		Tumor	1	0.98	0.99	0.997
BSIF	99.51%	Cyst	0.99	0.98	0.99	0.993
		Normal	0.99	0.99	0.99	0.991
		Stone	0.98	1	0.99	0.987
		Tumor	1	0.99	0.99	0.998
DCTNet	94.80%	Cyst	0.92	0.97	0.94	0.935
		Normal	0.97	0.94	0.96	0.974
		Stone	0.96	0.90	0.93	0.963
		Tumor	0.95	0.98	0.96	0.971
DLNet	98.36%	Cyst	0.98	0.98	0.98	0.985
		Normal	0.96	0.98	0.97	0.978
		Stone	1	0.99	0.99	0.995
		Tumor	1	0.98	0.99	0.997
MDFNet	99%	Cyst	0.99	1	0.99	0.995
		Normal	0.98	1	0.99	0.990
		Stone	1	0.99	0.99	0.998
		Tumor	1	0.99	0.99	0.998

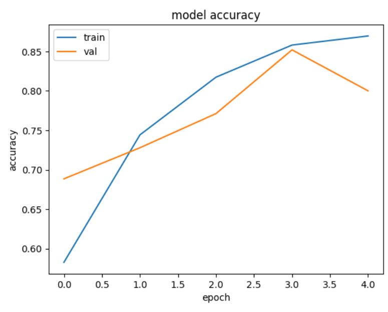
Figures 3.1, 3.2 summarizes the training and loss curves, demonstrating the improvement in classification performance of the CNN and Transformer models with increasing epochs.



Accuracy for VGG-16

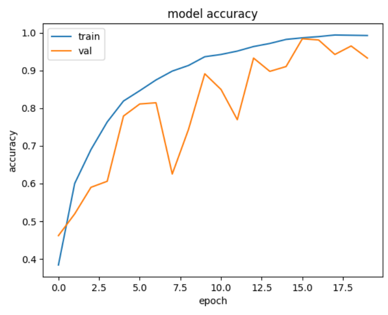


Accuracy for RESNet-50

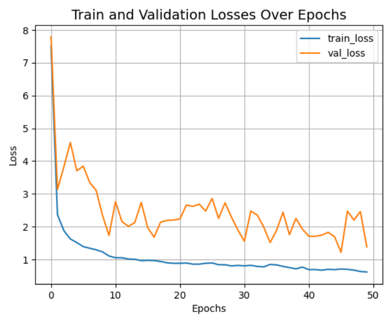


Accuracy for Inception-V3

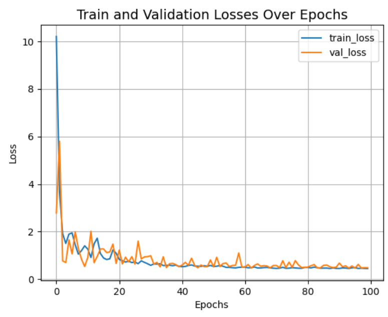
Figure 3.1: Accuracy curves for CNN Based Models Used On Original Dataset.



Accuracy for CCT



Loss for EANet

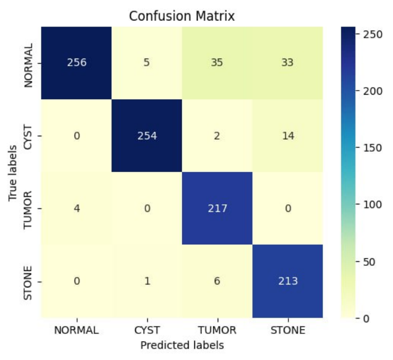


Loss for SWIN

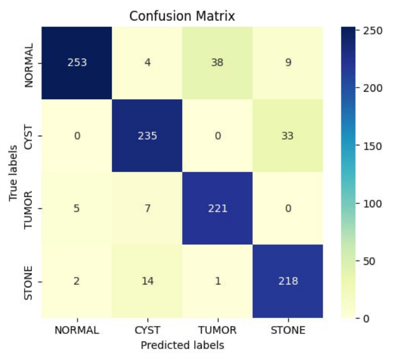
Figure 3.2: Accuracy and Loss curves for Transformer Based Models Used On Original Dataset.

Figures 3.3, 3.4, 3.5, and 3.6 shows the normalized Confusion Matrices for Transfer and Transformer, Lightweight based models, and Handcrafted consecutively.

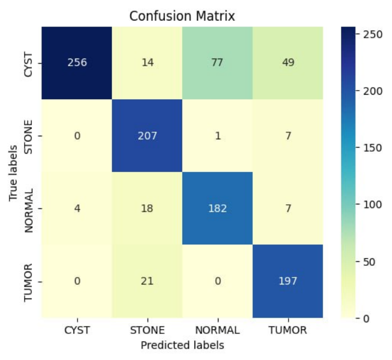
Figures 3.7, 3.8, 3.9, 3.10 provides the ROC curves for Transfer, Transformer, Lightweight based models, and Handcrafted consecutively.



Confusion Matrix for VGG-16

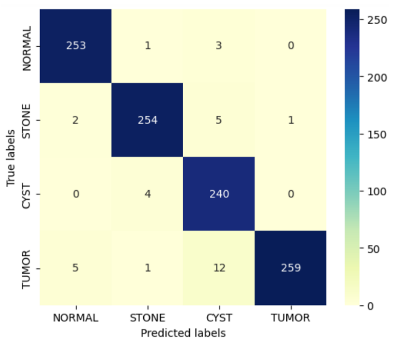


Confusion Matrix for RESNet-50

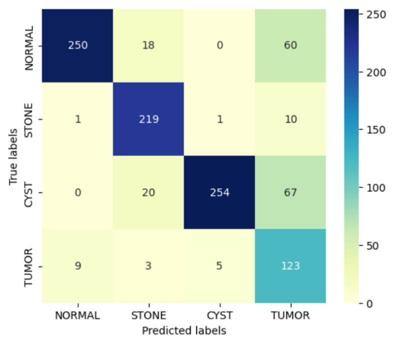


Confusion Matrix for Inception-V3

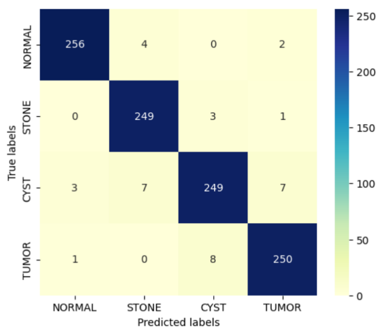
Figure 3.3: Confusion Matrix for CNN Based Models Used On Original Dataset.



Confusion Matrix for CCT

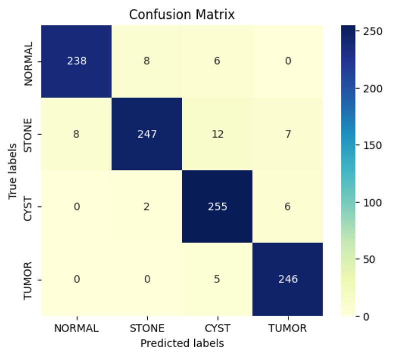


Confusion Matrix for EANet

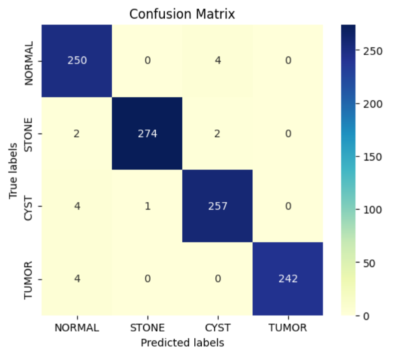


Confusion Matrix for SWIN

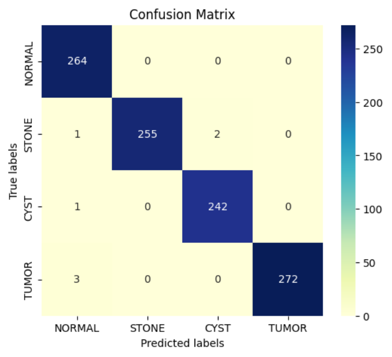
Figure 3.4: Confusion Matrix for Transformer Based Models Used On Original Dataset.



Confusion Matrix for DCT-Net

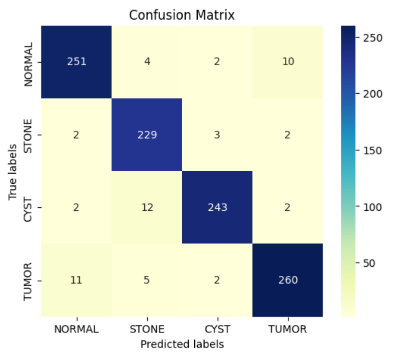


Confusion Matrix for DLNet

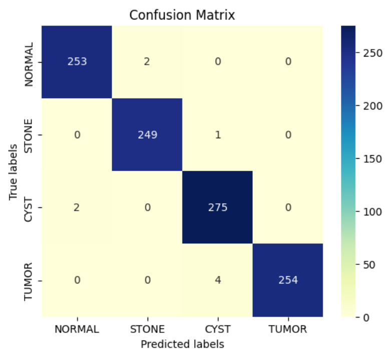


Confusion Matrix for MDFNet

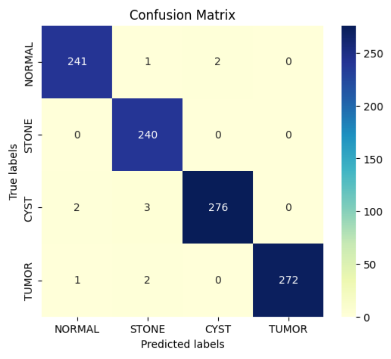
Figure 3.5: Confusion Matrix for Lightweight neural networks Based Models Used On Original Dataset.



Confusion Matrix for LBP

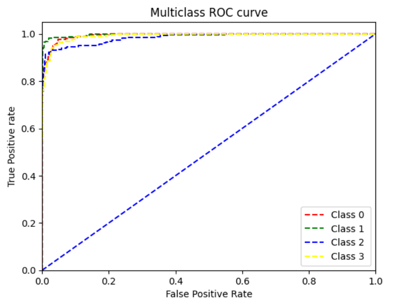


Confusion Matrix for HOG

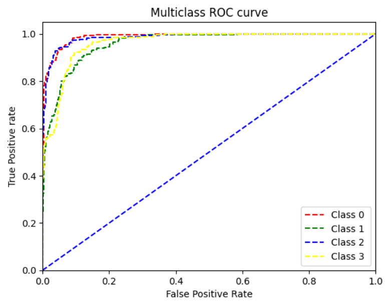


Confusion Matrix for BSIF

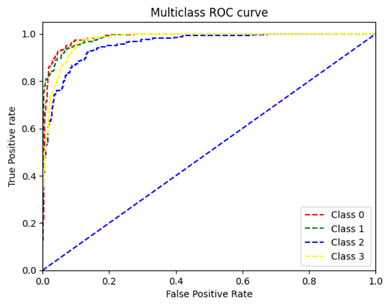
Figure 3.6: Confusion Matrix for Handcrafted features Based Models Used On Original Dataset.



Roc Curve for VGG-16

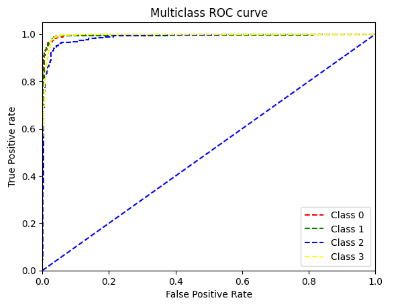


Roc Curve for RESNet-50

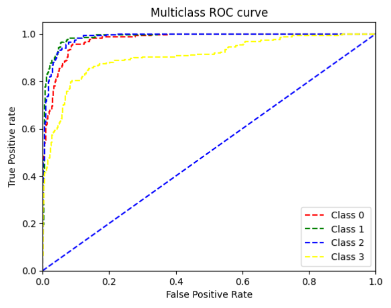


Roc Curve for Inception-v3

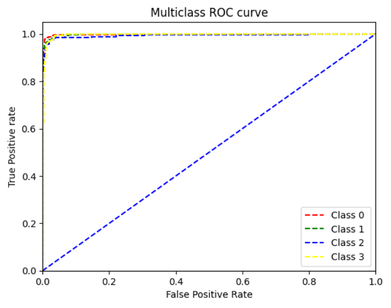
Figure 3.7: Roc Curves for CNN Based Models Used On Original Dataset.



Roc Curve for CCT

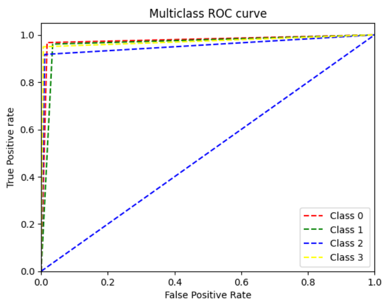


Roc Curve for EANet

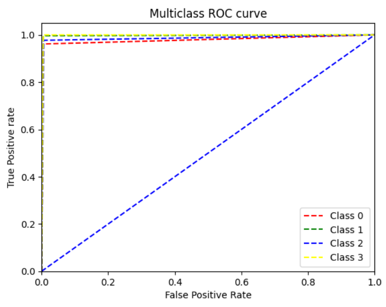


Roc Curve for SWIN

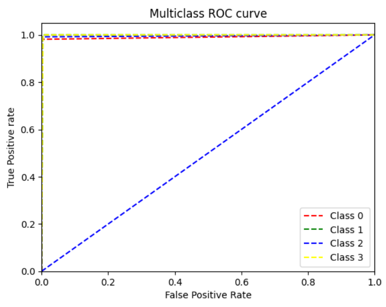
Figure 3.8: Roc Curves for Transformer Based Models Used On Original Dataset.



Roc Curve for DCTNET

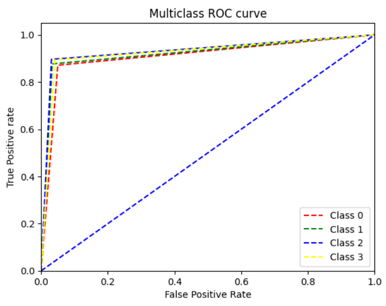


Roc Curve for DLNET

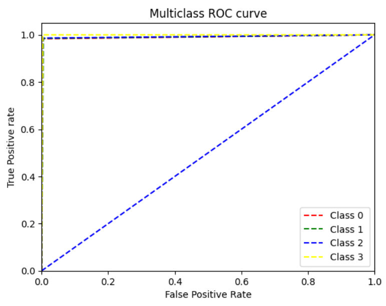


Roc Curve for MDFNET

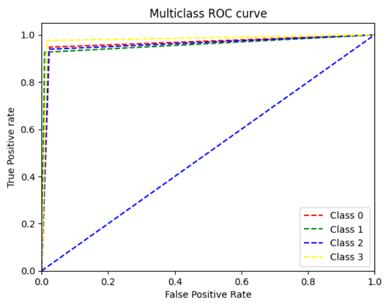
Figure 3.9: Roc Curves for Lightweight neural networks Based Models Used On Oroginal Dataset.



Roc Curve for LBP



Roc Curve for HOG



Roc Curve for BSIF

Figure 3.10: Roc Curves for Handcrafted features Based Models Used On Original Dataset.

Experiment 2: performance using the Blured dataset

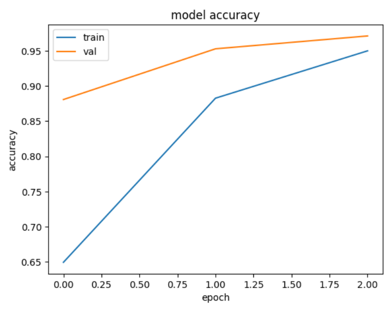
Upon analyzing the results presented in Table 3.2, our comparative analysis revealed varied performance among the different models on the Blurred dataset. The EANet model exhibited lower performance with an accuracy of 86.63%, while Inception-v3 and Resnet-50 demonstrated moderate performance with accuracies of 93.56% and 94.23% respectively. In contrast, CCT, VGG16, and Swin Transformers displayed higher accuracy rates of 98.75%, 98.08%, and 96.35% respectively. Interestingly, the handcrafted features LBP and HOG outperformed all other models, achieving accuracies of 99%. This superior performance can be attributed to the simplicity of the images in our dataset, which facilitated easier processing using traditional feature extraction techniques. It's worth noting that the amount of blur added to the dataset was relatively small. Additionally, the lightweight models achieved a remarkable accuracy of 99%.

The results presented in Table 3.2 provide insights into the performance of various models on the Blurred dataset across different evaluation metrics. The VGG-16 CNN model demonstrates reasonably good recall for detecting cyst, normal, stone, and tumor classes, with recall values of 0.97, 1, 0.99, and 0.96, respectively. Notably, it achieves a recall of 1 specifically for kidney normal classes, while the CCT model performs well in detecting cyst and stone classes with recall values of 1 for both. The MDFNet model exhibits perfect recall for the normal class, and the BSIF handcrafted feature achieves a recall of 1 specifically for the stone class. Recall values differ among various CNN models for the remaining classes, with Resnet-50, Inception-v3, EANet, SWIN, LBP, DCTNet, and DLNet showing varying performance. In terms of precision, VGG16 outperforms Inception V3 and Resnet50 in the transfer-based approach, with precision values of 1, 0.96, 0.98, and 0.99 for the respective classes. CCT stands out in the transformer-based approach with superior precision values compared to EANet and SWIN. BSIF exhibits high precision values in the handcrafted features-based approach, while DLNet surpasses DCTNet and MDFNet in the lightweight-based approach. F1 scores show that VGG-16 achieves the highest scores for cyst, normal, stone, and tumor classes, followed by CCT in the transformer-based approach. HOG demonstrates impressive F1 scores in the handcrafted feature-based approach, and DCTNet stands out in the lightweight-based approach. Examining the AUC values in Table 3.2, it is evident that all models perform well, with high AUC values close to 1 when diagnosing Kidney Cyst, Normal, Stone, and Tumor categories in the Blurred dataset.

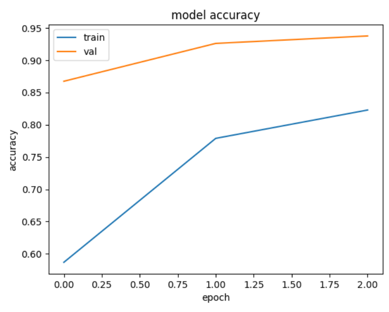
Table 3.2: The performance measures for all the models on the BLUR dataset

Models	Accuracy	Class	Precision	Recall(Sensitivity)	F1 Score	AUC
VGG16	98.08%	Cyst	1	0.97	0.98	0.999
		Normal	0.96	1	0.98	0.999
		Stone	0.98	0.99	0.99	0.999
		Tumor	0.99	0.96	0.98	0.999
Resnet50	94.23%	Cyst	0.88	1	0.93	0.999
		Normal	0.98	0.91	0.95	0.998
		Stone	0.98	0.88	0.93	0.995
		Tumor	0.93	1	0.96	0.999
Inception v3	93.56%	Cyst	0.96	0.94	0.95	0.993
		Normal	0.91	0.91	0.91	0.992
		Stone	0.91	0.96	0.93	0.994
		Tumor	0.97	0.94	0.95	0.997
CCT	98.75%	Cyst	0.97	1	0.99	0.993
		Normal	1	0.97	0.99	0.999
		Stone	0.99	1	1	0.999
		Tumor	0.98	0.98	0.98	0.995
EANet	86.63%	Cyst	0.97	0.89	0.93	0.99
		Normal	0.95	0.76	0.85	0.988
		Stone	0.98	0.91	0.94	0.996
		Tumor	0.57	0.95	0.71	0.953
SWIN	96.35%	Cyst	0.93	0.99	0.96	0.996
		Normal	0.99	0.94	0.97	0.998
		Stone	0.97	0.98	0.97	0.999
		Tumor	0.96	0.95	0.95	0.995
LBP	99%	Cyst	0.98	0.96	0.97	0.979
		Normal	0.95	0.97	0.96	0.970
		Stone	0.94	0.98	0.96	0.964
		Tumor	0.98	0.93	0.96	0.979
HOG	99%	Cyst	0.99	0.98	0.99	0.993
		Normal	0.98	0.99	0.99	0.989
		Stone	0.99	0.99	0.99	0.993
		Tumor	1	0.99	1	0.998
BSIF	98.36%	Cyst	1	0.98	0.99	0.994
		Normal	0.97	0.99	0.98	0.982
		Stone	0.98	1	0.99	0.989
		Tumor	0.99	0.97	0.98	0.989
DCNet	99%	Cyst	0.99	0.99	0.99	0.994
		Normal	0.99	1	1	0.996
		Stone	1	1	1	0.997
		Tumor	1	0.99	1	0.998
DLNet	99%	Cyst	0.98	1	0.99	0.991
		Normal	1	0.99	1	0.998
		Stone	1	0.99	0.99	0.998
		Tumor	1	1	1	1
MDFNet	99%	Cyst	1	0.99	0.99	0.996
		Normal	0.98	1	0.99	0.99
		Stone	1	0.99	0.99	0.996
		Tumor	1	0.99	0.99	0.996

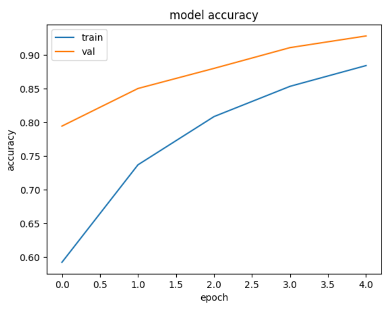
Figures 3.11, 3.12 summarizes the training and loss curves, demonstrating the improvement in classification performance of the CNN and Transformer models with increasing epochs.



Accuracy for VGG-16

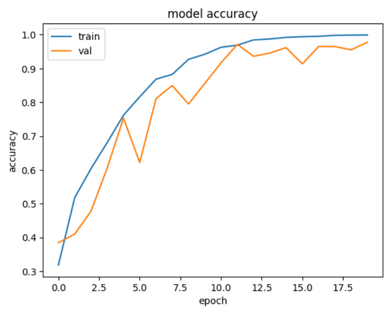


Accuracy for RESNet-50

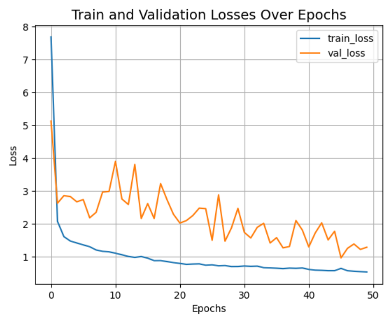


Accuracy for Inception-V3

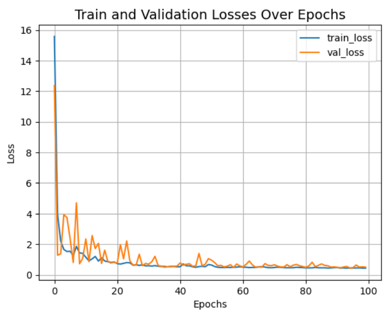
Figure 3.11: Accuracy curves for CNN Based Models Used On Blured Dataset.



Accuracy for CCT



Loss for EANet

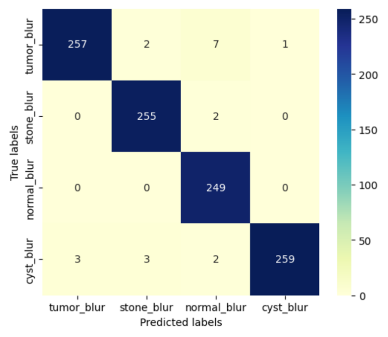


Loss for SWIN

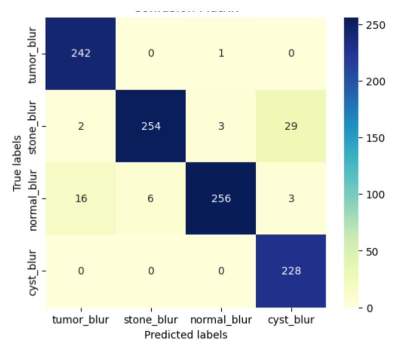
Figure 3.12: Accuracy and Loss curves for Transformer Based Models Used On Blured Dataset.

Figures 3.13, 3.14, 3.15, and 3.16 shows the normalized Confusion Matrices for Transfer and Transformer, Lightweight based models, and Handcrafted consecutively.

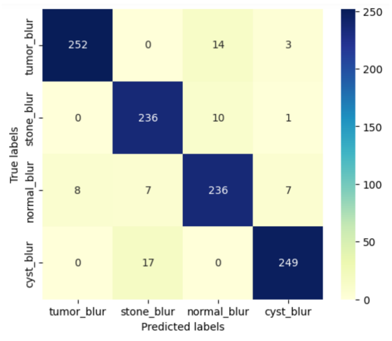
Figures 3.17, 3.18, 3.19, 3.20 provides the ROC curves for Transfer, Transformer, Lightweight based models, and Handcrafted consecutively.



Confusion Matrix for VGG-16

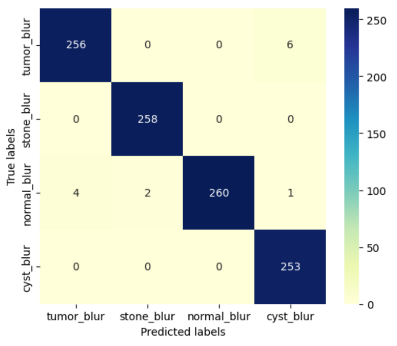


Confusion Matrix for RESNet-50

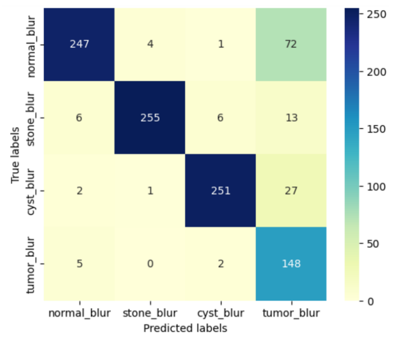


Confusion Matrix for Inception-V3

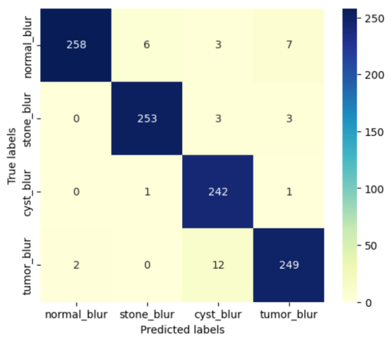
Figure 3.13: Confusion Matrix for CNN Based Models Used On Blured Dataset.



Confusion Matrix for CCT

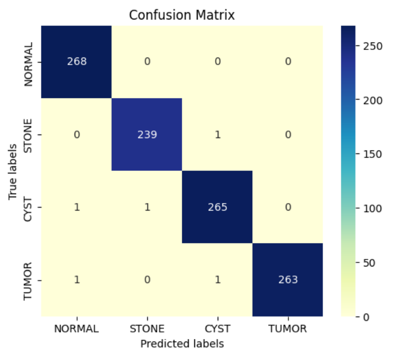


Confusion Matrix for EANet

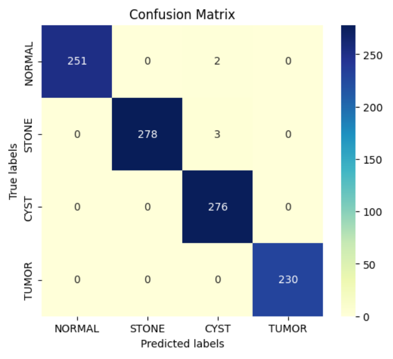


Confusion Matrix for SWIN

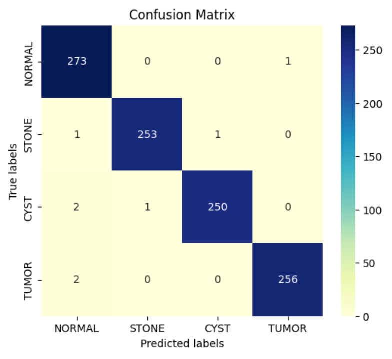
Figure 3.14: Confusion Matrix for Transformer Based Models Used On Blured Dataset.



Confusion Matrix for DCT-Net

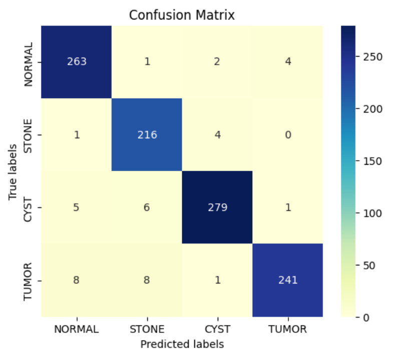


Confusion Matrix for DLNet

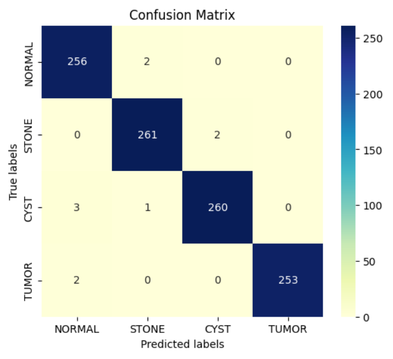


Confusion Matrix for MDFNet

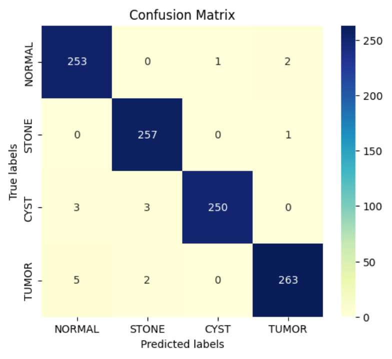
Figure 3.15: Confusion Matrix for Lightweight neural networks Based Models Used On Blured Dataset.



Confusion Matrix for LBP

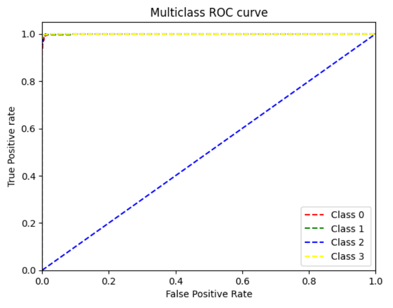


Confusion Matrix for HOG

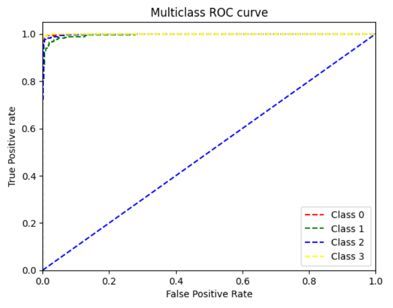


Confusion Matrix for BSIF

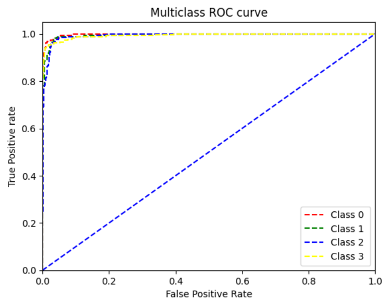
Figure 3.16: Confusion Matrix for Handcrafted features Based Models Used On Blured Dataset.



Roc Curve for VGG-16

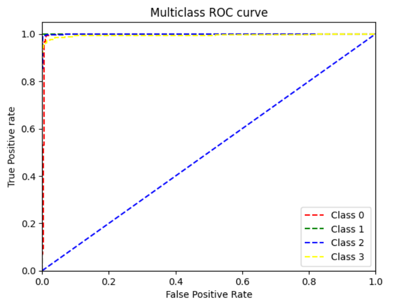


Roc Curve for RESNet-50

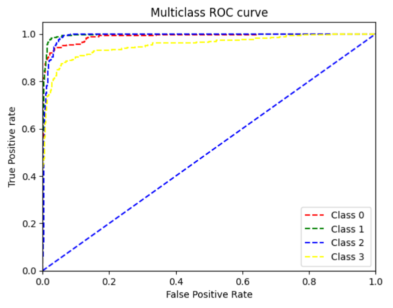


Roc Curve for Inception-v3

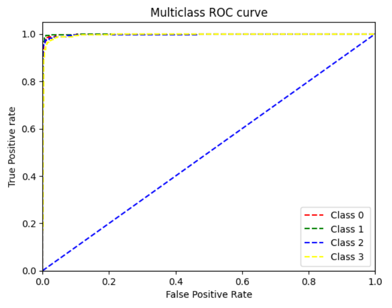
Figure 3.17: Roc Curves for CNN Based Models Used On Blured Dataset.



Roc Curve for CCT

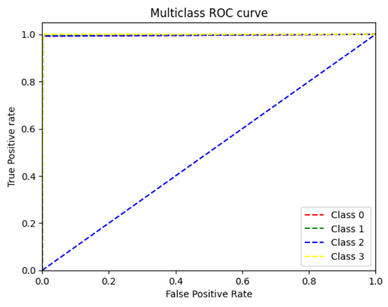


Roc Curve for EANet

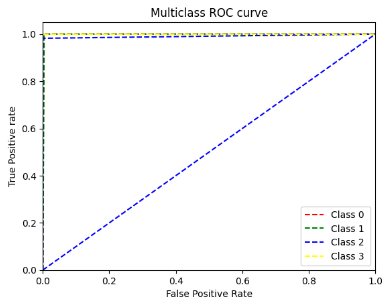


Roc Curve for SWIN

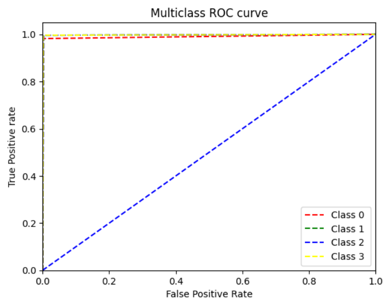
Figure 3.18: Roc Curves for Transformer Based Models Used On Blured Dataset.



Roc Curve for DCTNET

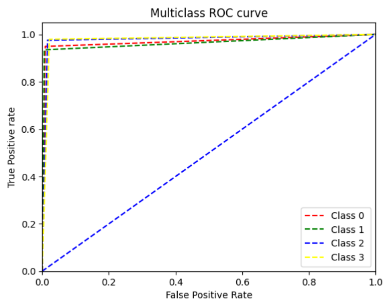


Roc Curve for DLNET

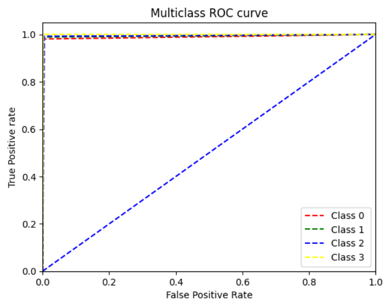


Roc Curve for MDFNET

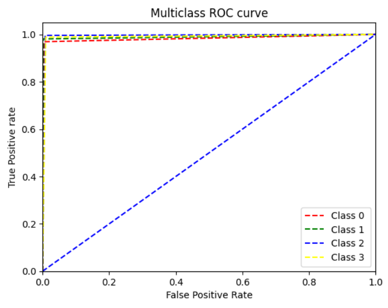
Figure 3.19: Roc Curves for Lightweight neural networks Based Models Used On Blured Dataset.



Roc Curve for LBP



Roc Curve for HOG



Roc Curve for BSIF

Figure 3.20: Roc Curves for Handcrafted features Based Models Used On Blured Dataset.

Experiment 3: performance using the Illumination Increase dataset

According to the results presented in Table 3.3, our comparative analysis revealed varying performance among the different models with respect to our illumination increase dataset. The SWIN and EANet models exhibited lower performance, achieving accuracies of 47.88% and 52.5% respectively. Inception-v3, VGG-16, and CCT demonstrated moderate performance with accuracies of 82.60%, 87.50%, and 88.65% respectively. On the other hand, the RESNet-50 Transformers displayed a higher accuracy of 90.29%. Interestingly, the handcrafted features, namely LBP, HOG, and BSIF, outperformed all other models with accuracies of 98.26%, 99%, and 99% respectively.

In situations where the illumination of an image increases or decreases, handcrafted features can still achieve high accuracy. Let's focus on the scenario where the image becomes very bright. When we apply a technique called Local Binary Patterns (LBP) to these bright images, something interesting happens.

LBP calculates binary values based on comparing the intensity of a center pixel with its neighboring pixels. The resulting binary values describe the patterns present in the image. Now, because the image has become very bright, the intensity values of the pixels have increased, including both the center pixel and its neighbors.

However, despite the increase in intensity, the relative differences between the center pixel and its neighbors remain relatively unchanged. This means that the order of the neighbors in relation to the center pixel doesn't change significantly, resulting in similar binary values as the original image.

The low accuracy achieved by the transformer model can be attributed to the illumination changes introduced in the original dataset. When certain zones of the images become overly bright and appear predominantly white, it can pose challenges for the transformer's attention mechanism.

According to the information presented in Table 3.3, the RESNet-50 CNN model demonstrates reasonably good recall for detecting cyst, normal, stone, and tumor classes, with recall values of 0.84, 0.92, 0.99, and 0.89 respectively. The CCT model performs well in detecting cyst and stone class images, with recall values of 0.78, 0.95, 0.95, and 0.90 for the respective classes. Among the handcrafted features, HOG achieves a recall of 1 specifically for both the normal and stone classes.

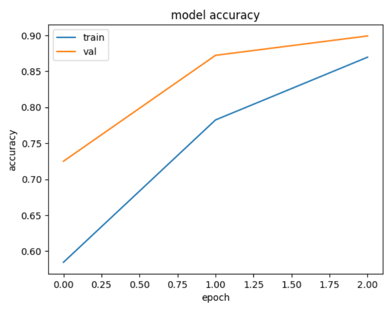
When considering precision, RESNet-50 exhibits higher precision values of 0.93, 0.87, 0.85, and 0.97 for the respective classes in the transfer-based approach. CCT stands out with superior precision values of 0.93, 0.88, 0.89, and 0.85 in the transformer-based approach. HOG demonstrates high precision values of 1, 0.97, 1, and 1 for the respective classes in the handcrafted features-based approach. In the lightweight-based approach, all DCNet, DLNet, and MDFNet achieve high precision.

In terms of F1 scores, RESNet-50 achieves the highest scores among the cyst, normal, stone, and tumor classes, with values of 0.88, 0.89, 0.92, and 0.92 respectively. CCT demonstrates strong F1 scores of 0.85, 0.91, 0.92, and 0.88 in the transformer-based approach. The handcrafted feature-based approaches, including LBP, HOG, and BSIF, exhibit impressive F1 scores close to 1, as do the lightweight networks.

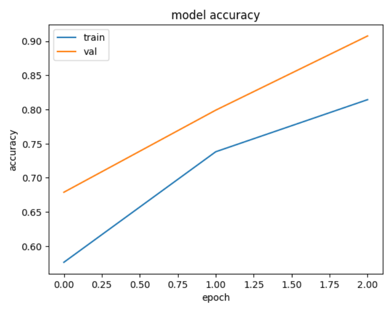
Table 3.3: The performance measures for all the models on the ILLUMINATION INCREASE dataset

Models	Accuracy	Class	Precision	Recall(Sensitivity)	F1 Score	AUC
VGG16	87.50%	Cyst	0.96	0.82	0.88	0.989
		Normal	0.81	0.90	0.85	0.986
		Stone	0.78	0.94	0.86	0.981
		Tumor	0.94	0.86	0.90	0.991
Resnet	90.29%	Cyst	0.93	0.84	0.88	0.983
		Normal	0.87	0.92	0.89	0.986
		Stone	0.85	0.99	0.92	0.990
		Tumor	0.97	0.89	0.92	0.994
Inception v3	82.60%	Cyst	0.96	0.62	0.75	0.966
		Normal	0.82	0.86	0.84	0.969
		Stone	0.55	0.83	0.66	0.926
		Tumor	0.70	0.84	0.77	0.963
CCT	88.65%	Cyst	0.93	0.78	0.85	0.955
		Normal	0.88	0.95	0.91	0.990
		Stone	0.89	0.95	0.92	0.987
		Tumor	0.85	0.90	0.88	0.971
EANet	52.5%	Cyst	0	0	0	0.591
		Normal	0.77	0.57	0.65	0.858
		Stone	0.77	0.52	0.62	0.841
		Tumor	0.57	0.49	0.52	0.741
SWIN	47.88%	Cyst	0.60	0.48	0.53	0.798
		Normal	0.59	0.56	0.57	0.851
		Stone	0.09	0.56	0.16	0.741
		Tumor	0.63	0.41	0.49	0.765
LBP	98.26%	Cyst	0.97	0.94	0.96	0.974
		Normal	0.93	0.99	0.96	0.962
		Stone	0.90	0.96	0.93	0.946
		Tumor	0.98	0.90	0.94	0.974
HOG	99%	Cyst	1	0.97	0.99	0.995
		Normal	0.97	1	0.98	0.983
		Stone	1	1	1	1
		Tumor	1	0.99	1	0.998
BSIF	99.58%	Cyst	1	0.99	0.99	0.996
		Normal	0.99	0.99	0.99	0.994
		Stone	0.98	1	0.99	0.991
		Tumor	0.99	0.98	0.99	0.991
DCNet	99%	Cyst	0.99	1	0.99	0.993
		Normal	0.99	1	0.99	0.992
		Stone	1	0.98	0.99	0.996
		Tumor	1	1	1	0.999
DLNet	99%	Cyst	1	0.99	1	0.998
		Normal	0.99	1	0.99	0.992
		Stone	1	1	1	0.999
		Tumor	1	1	1	0.999
MDFNet	99%	Cyst	1	1	1	0.998
		Normal	0.97	1	0.98	0.985
		Stone	1	1	1	0.999
		Tumor	1	0.97	0.98	0.994

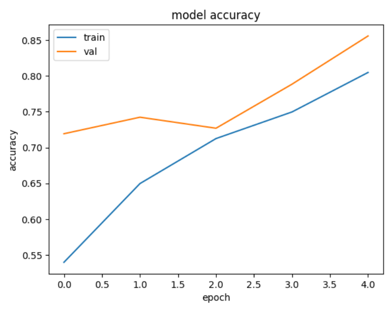
Figures 3.21, 3.22 summarizes the training and loss curves, demonstrating the improvement in classification performance of the CNN and Transformer models with increasing epochs.



Accuracy for VGG-16

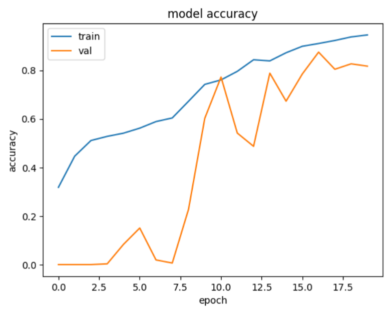


Accuracy for RESNet-50

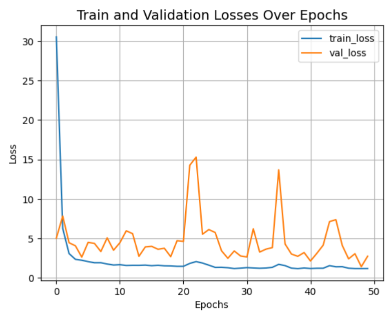


Accuracy for Inception-V3

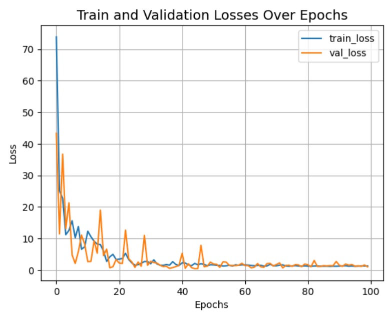
Figure 3.21: Accuracy curves for CNN Based Models Used On Illumination Increase Dataset.



Accuracy for CCT



Loss for EANet

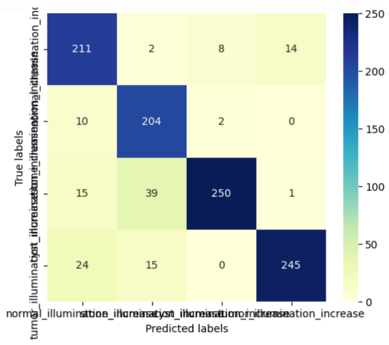


Loss for SWIN

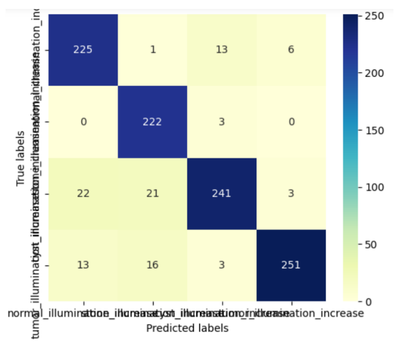
Figure 3.22: Accuracy and Loss curves for Transformer Based Models Used On Illumination Increase Dataset.

Figures 3.23, 3.24, 3.25, and 3.26 shows the normalized Confusion Matrices for Transfer and Transformer, Lightweight based models, and Handcrafted consecutively for illumination increase dataset.

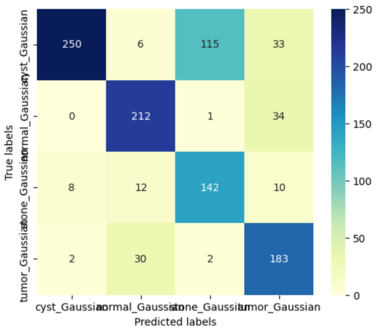
Figures 3.27, 3.28, 3.29, 3.30 provides the ROC curves for Transfer, Transformer, Lightweight based models, and Handcrafted consecutively for illumination increase dataset.



Confusion Matrix for VGG-16

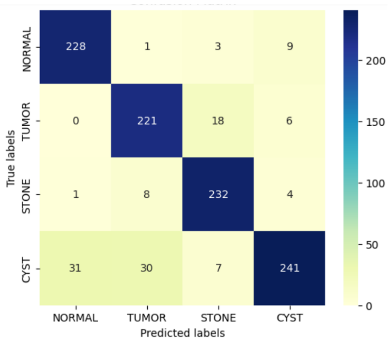


Confusion Matrix for RESNet-50

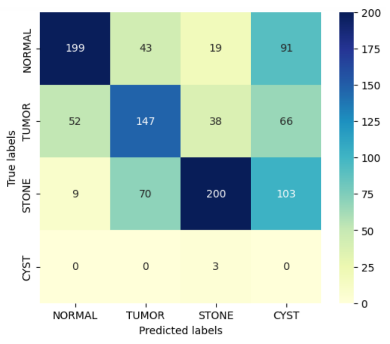


Confusion Matrix for Inception-V3

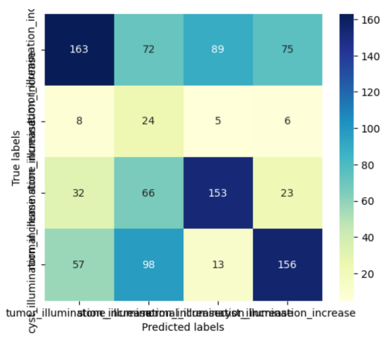
Figure 3.23: Confusion Matrix for CNN Based Models Used On Illumination Increase Dataset.



Confusion Matrix for CCT

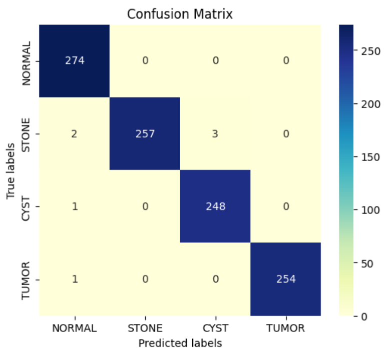


Confusion Matrix for EANet

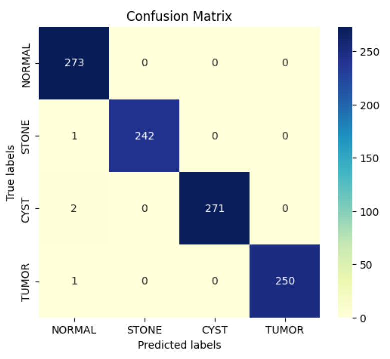


Confusion Matrix for SWIN

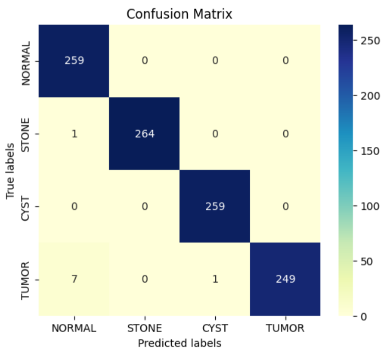
Figure 3.24: Confusion Matrix for Transformer Based Models Used On Illumination Increase Dataset.



Confusion Matrix for DCT-Net

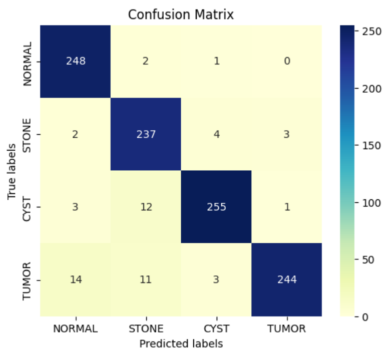


Confusion Matrix for DLNet

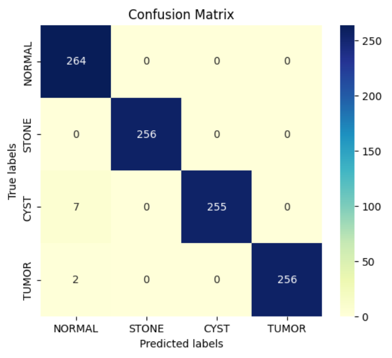


Confusion Matrix for MDFNet

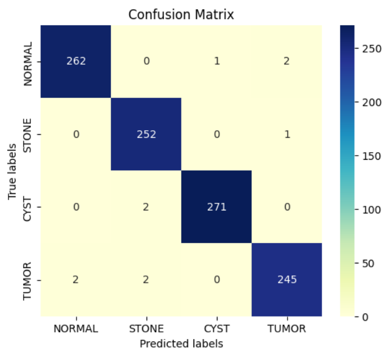
Figure 3.25: Confusion Matrix for Lightweight neural networks Based Models Used On Illumination Increase Dataset.



Confusion Matrix for LBP

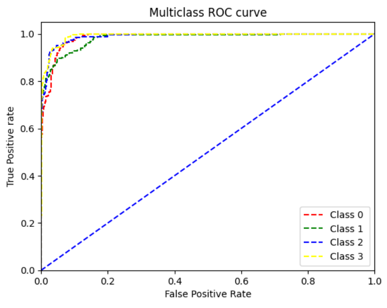


Confusion Matrix for HOG

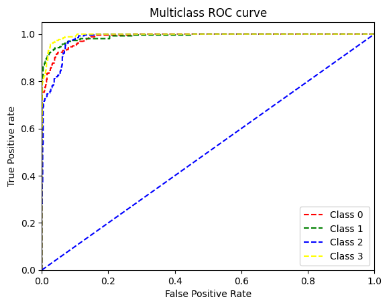


Confusion Matrix for BSIF

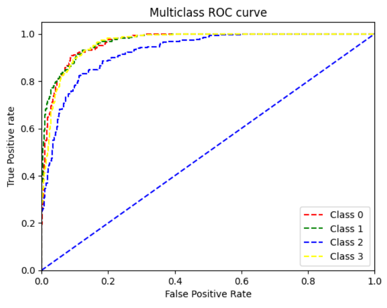
Figure 3.26: Confusion Matrix for Handcrafted features Based Models Used On Illumination Increase Dataset.



Roc Curve for VGG-16

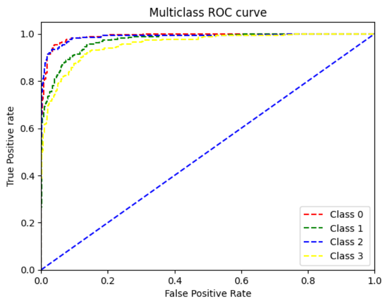


Roc Curve for RESNet-50

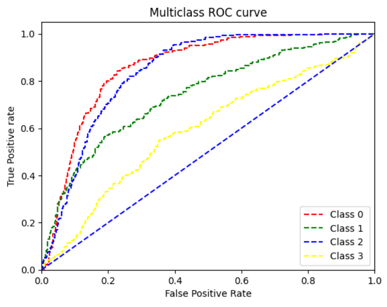


Roc Curve for Inception-v3

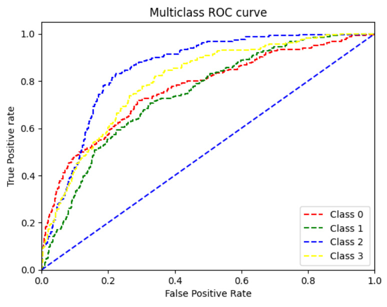
Figure 3.27: Roc Curves for CNN Based Models Used On Illumination Increase Dataset.



Roc Curve for CCT

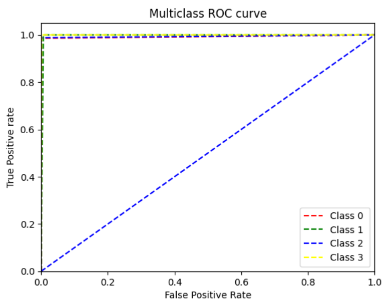


Roc Curve for EANet

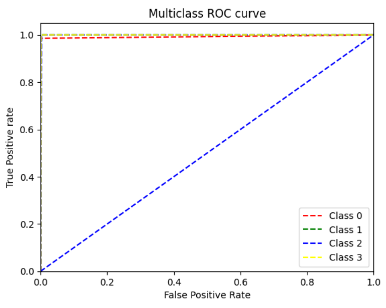


Roc Curve for SWIN

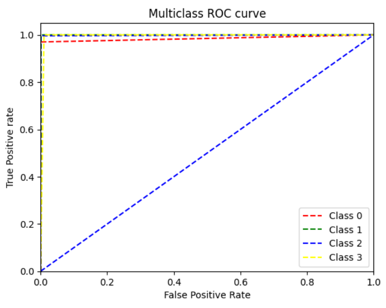
Figure 3.28: Roc Curves for Transformer Based Models Used On Illumination Increase Dataset.



Roc Curve for DCTNET

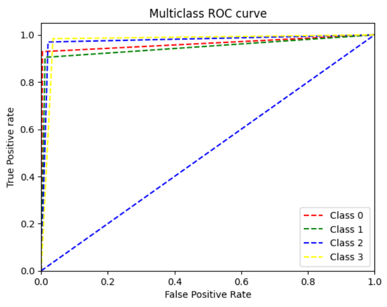


Roc Curve for DLNET

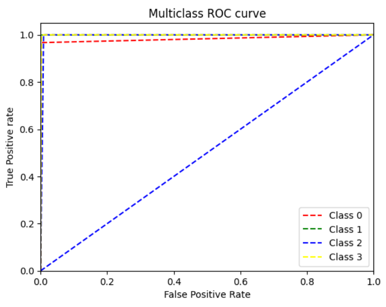


Roc Curve for MDFNET

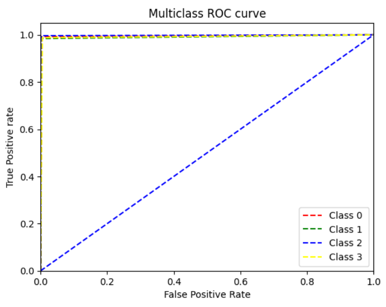
Figure 3.29: Roc Curves for Lightweight neural networks Based Models Used On Illumination Increase Dataset.



Roc Curve for LBP



Roc Curve for HOG



Roc Curve for BSIF

Figure 3.30: Roc Curves for Handcrafted features Based Models Used On Illumination Increase Dataset.

Experiment 4: performance using the Occlusion dataset

According to the results presented in Table 3.4, our comparative analysis revealed varying performance among the different models with respect to our occlusion dataset. The EANet model exhibited lower performance with an accuracy of 68.94%. Inception-v3 and RESNet-50 demonstrated moderate performance with accuracies of 72.88% and 81.35% respectively. On the other hand, VGG-16, CCT, and SWIN displayed higher accuracy of 93.46%, 93.37%, and 91.35% respectively.

Interestingly, the handcrafted features, namely LBP, HOG, and BSIF, outperformed all other models with accuracies of 95.86%, 99%, and 94.71% respectively. Similarly, the lightweight models achieved high accuracy with values of 99.03%, 98%, and 99% respectively.

The relatively lower accuracy achieved by the transformer model can be attributed to the presence of occlusion in the original dataset. The occlusion introduces challenges for the transformer's attention mechanism, as it struggles to effectively capture the relevant information when certain zones of the images are obscured or missing.

According to the information presented in Table 3.4, our analysis reveals varying performance among the different models with respect to the occlusion dataset. The VGG-16 CNN model demonstrates reasonably good recall for detecting cyst, normal, stone, and tumor classes, with recall values of 0.91, 0.96, 0.96, and 0.91, respectively. The CCT model performs well in detecting cyst and stone class images, with recalls of 0.96 and 0.97, and also achieves a recall of 0.95 for the normal class and 0.87 for the tumor class.

The lightweight networks exhibit perfect recall, indicating their strong performance in detecting occluded images. Among the handcrafted features, HOG achieves high recall values close to 1 for all classes.

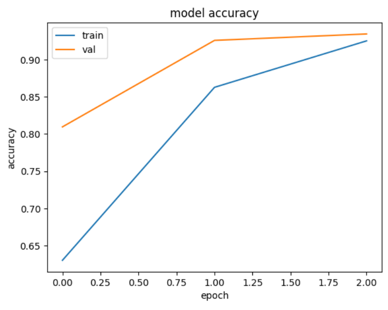
When considering precision, VGG-16 exhibits higher precision values of 0.97, 0.94, 0.92, and 0.90 for the respective classes. In the transformer-based approach, CCT stands out with superior precision values of 0.87, 0.95, 0.96, and 0.95 compared to EANet and SWIN. HOG demonstrates precision values of 0.99, 0.98, 0.98, and 1 for the respective classes in the handcrafted features-based approach. In the lightweight-based approach, all DCTNet, DLNet, and MDFNet achieve high precision.

In terms of F1 scores, VGG-16 achieves the highest scores among the cyst, normal, stone, and tumor classes, with values of 0.94, 0.95, 0.94, and 0.91, respectively. CCT demonstrates strong F1 scores of 0.91, 0.95, 0.96, and 0.91 in the transformer-based approach. The lightweight-based approaches, including DCTNet, DLNet, and MDFNet, exhibit impressive F1 scores close to 1. Similarly, the handcrafted features also achieve high F1 scores.

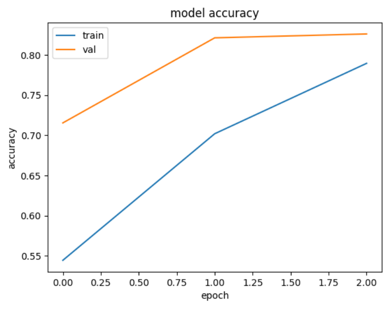
Table 3.4: The performance measures for all the models on the OCCLUSION dataset

Models	Accuracy	Class	Precision	Recall(Sensitivity)	F1 Score	AUC
VGG16	93.46%	Cyst	0.97	0.91	0.94	0.996
		Normal	0.94	0.96	0.95	0.997
		Stone	0.92	0.96	0.94	0.995
		Tumor	0.90	0.91	0.91	0.989
Resnet	81.35%	Cyst	0.94	0.82	0.88	0.988
		Normal	0.55	0.95	0.70	0.964
		Stone	0.81	0.78	0.79	0.959
		Tumor	0.96	0.77	0.85	0.985
Inception v3	72.88%	Cyst	0.92	0.64	0.75	0961
		Normal	0.88	0.68	077	0.950
		Stone	0.56	0.96	0.71	0.968
		Tumor	0.55	0.83	0.66	O.950
CCT	93.37%	Cyst	0.87	0.96	0.91	0.978
		Normal	0.95	0.95	0.95	0.987
		Stone	0.96	0.97	0.96	0.993
		Tumor	0.95	0.87	0.91	0.977
EANet	68.94%	Cyst	0.93	0.76	0.83	0.968
		Normal	0.91	0.56	0.70	0.899
		Stone	0.92	0.79	0.85	0.979
		Tumor	0	0	0	0.814
SWIN	91.35%	Cyst	0.95	0.90	0.93	0.992
		Normal	0.94	0.90	0.92	0.990
		Stone	0.97	0.88	0.92	0.994
		Tumor	0.79	1	0.88	0.993
LBP	95.86%	Cyst	0.90	0.91	0.90	0.932
		Normal	0.87	0.85	0.86	0.910
		Stone	0.88	0.89	0.89	0.921
		Tumor	0.89	0.89	0.89	0.926
HOG	99%	Cyst	0.99	0.99	0.99	0.991
		Normal	0.98	0.99	0.99	0.990
		Stone	0.98	0.98	0.98	0.989
		Tumor	1	0.99	0.99	0.998
BSIF	94.71%	Cyst	0.94	0.94	0.94	0.958
		Normal	0.95	0.93	0.94	0.962
		Stone	0.93	0.97	0.95	0.958
		Tumor	0.98	0.95	0.96	0.979
DCNet	99.03%	Cyst	1	0.99	0.99	0.996
		Normal	0.98	1	0.99	0.987
		Stone	0.99	0.99	0.99	0.993
		Tumor	1	0.98	0.99	0.997
DLNet	98%	Cyst	0.99	0.97	0.98	0.990
		Normal	0.97	1	0.98	0.984
		Stone	0.98	0.99	0.98	0.986
		Tumor	1	0.99	0.99	0.997
MDFNet	99%	Cyst	0.99	0.98	0.98	0.989
		Normal	0.99	0.99	0.99	0.992
		Stone	0.99	1	0.99	0.993
		Tumor	1	0.99	0.99	0.996

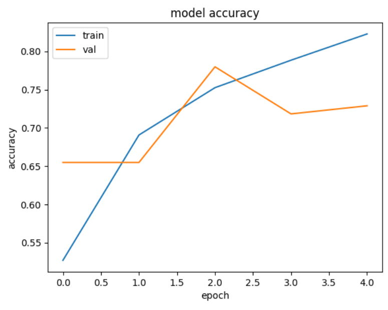
Figures 3.31, 3.32 summarizes the training and loss curves, demonstrating the improvement in classification performance of the CNN and Transformer models with increasing epochs.



Accuracy for VGG-16

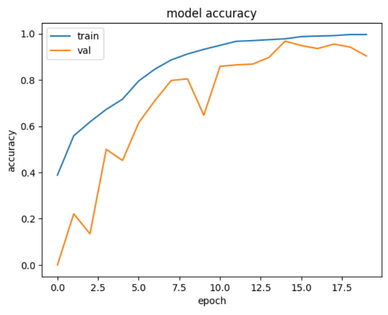


Accuracy for RESNet-50

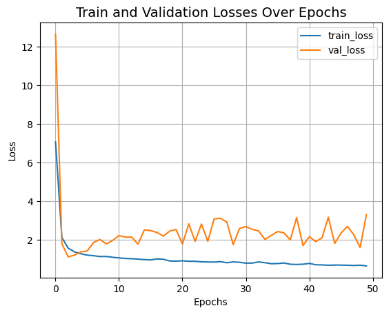


Accuracy for Inception-V3

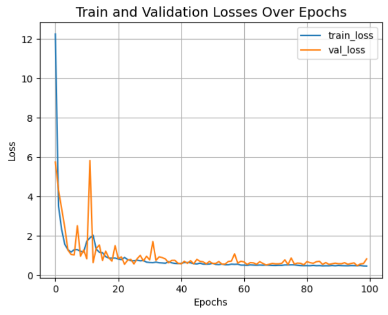
Figure 3.31: Accuracy curves for CNN Based Models Used On Occlusion Dataset.



Accuracy for CCT



Loss for EANet

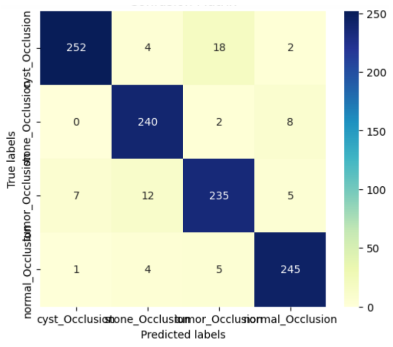


Loss for SWIN

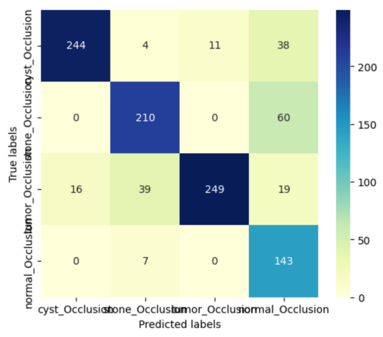
Figure 3.32: Accuracy and Loss curves for Transformer Based Models Used On Occlusion Dataset.

Figures 3.33, 3.34, 3.35, and 3.36 shows the normalized Confusion Matrices for Transfer and Transformer, Lightweight based models, and Handcrafted consecutively for occlusion dataset.

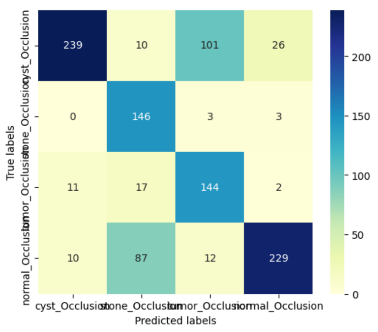
Figures 3.37, 3.38, 3.39, 3.30 provides the ROC curves for Transfer, Transformer, Lightweight based models, and Handcrafted consecutively for occlusion dataset.



Confusion Matrix for VGG-16

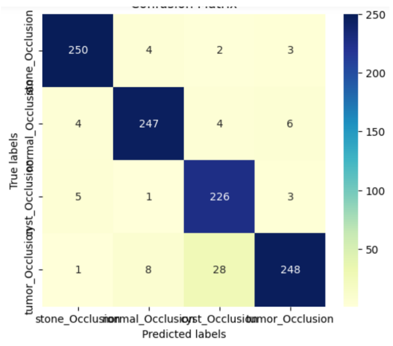


Confusion Matrix for RESNet-50

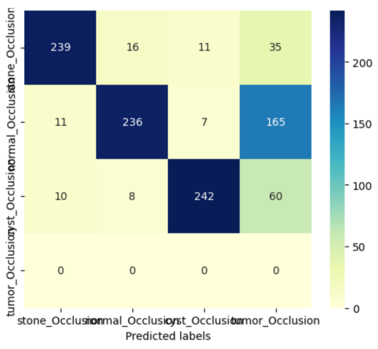


Confusion Matrix for Inception-V3

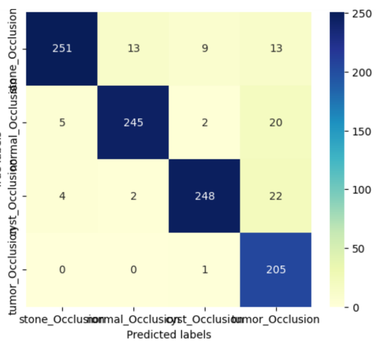
Figure 3.33: Confusion Matrix for CNN Based Models Used On Occlusion Dataset.



Confusion Matrix for CCT

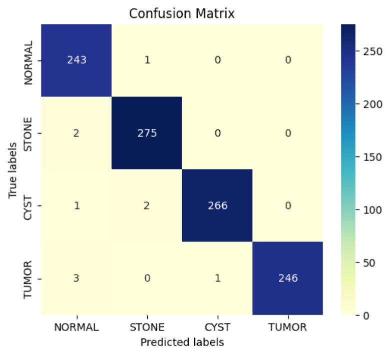


Confusion Matrix for EANet

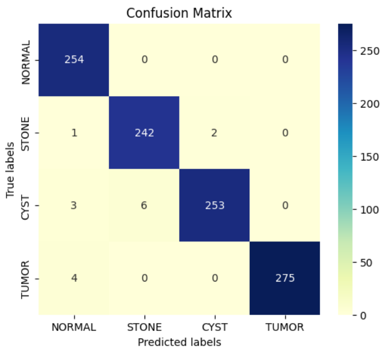


Confusion Matrix for SWIN

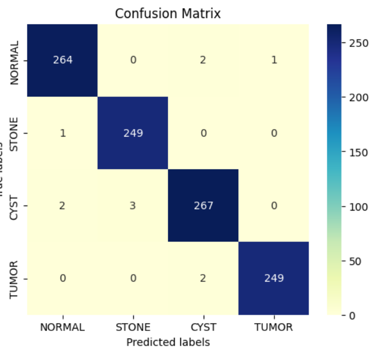
Figure 3.34: Confusion Matrix for Transformer Based Models Used On Occlusion Dataset.



Confusion Matrix for DCT-Net

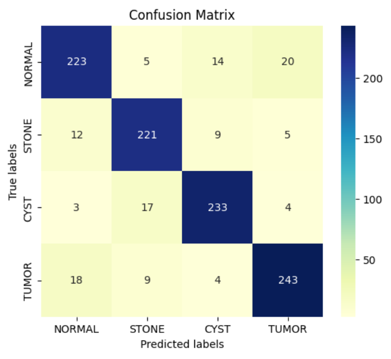


Confusion Matrix for DLNet

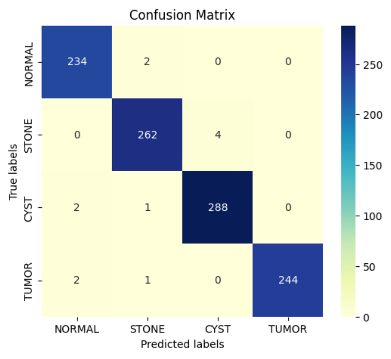


Confusion Matrix for MDFNet

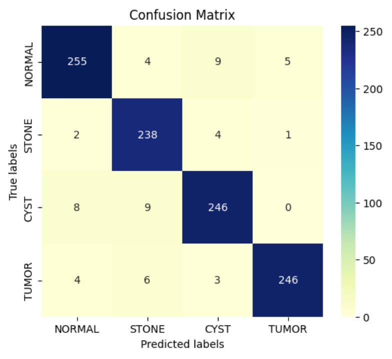
Figure 3.35: Confusion Matrix for Lightweight neural networks Based Models Used On Occlusion Dataset.



Confusion Matrix for LBP

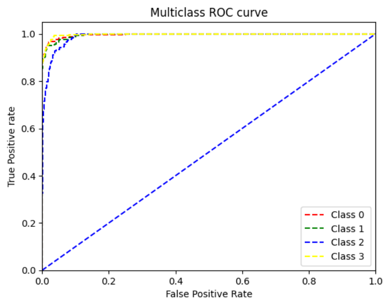


Confusion Matrix for HOG

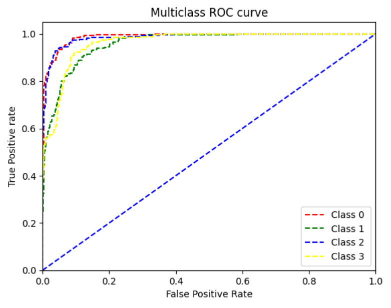


Confusion Matrix for BSIF

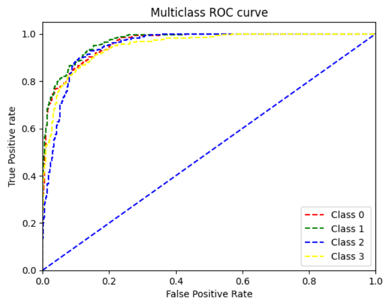
Figure 3.36: Confusion Matrix for Handcrafted features Based Models Used On Occlusion Dataset.



Roc Curve for VGG-16

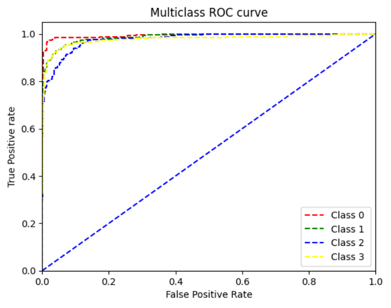


Roc Curve for RESNet-50

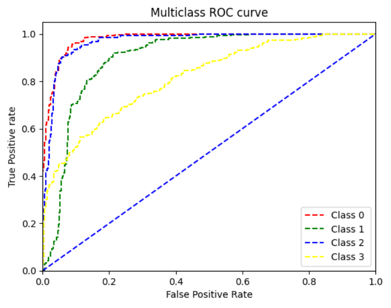


Roc Curve for Inception-v3

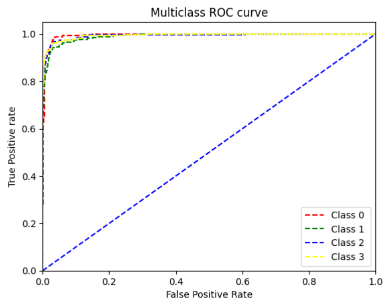
Figure 3.37: Roc Curves for CNN Based Models Used On Occlusion Dataset.



Roc Curve for CCT

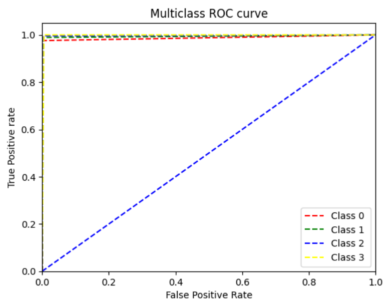


Roc Curve for EANet

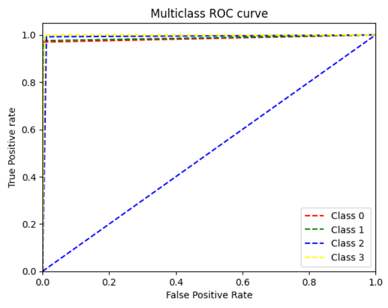


Roc Curve for SWIN

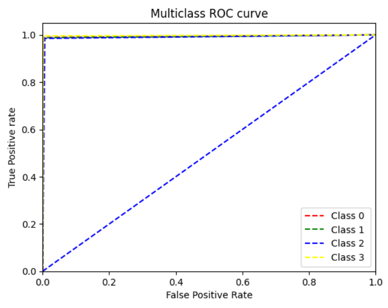
Figure 3.38: Roc Curves for Transformer Based Models Used On Occlusion Dataset.



Roc Curve for DCTNET

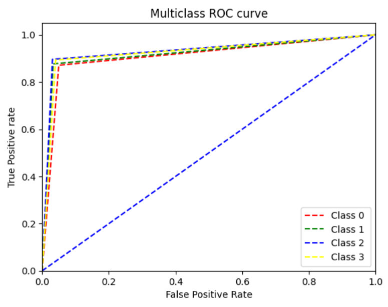


Roc Curve for DLNET

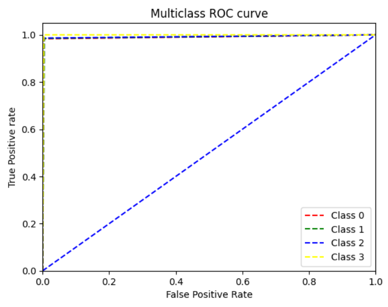


Roc Curve for MDFNET

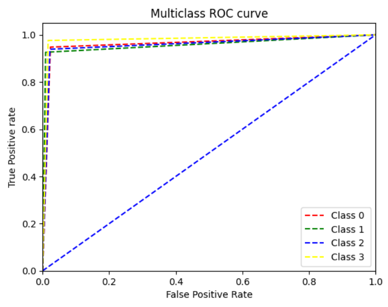
Figure 3.39: Roc Curves for Lightweight neural networks Based Models Used On Occlusion Dataset.



Roc Curve for LBP



Roc Curve for HOG



Roc Curve for BSIF

Figure 3.40: Roc Curves for Handcrafted features Based Models Used On Occlusion Dataset.

Experiment 5: performance using the Gaussian noise dataset

According to the results presented in Table 3.5, our comparative analysis revealed varying performance among the different models with respect to our Gaussian dataset. The EANet and SWIN models exhibited lower performance, achieving accuracies of 54.62% and 65.29%, respectively. Inception-v3, RESNet-50, and CCT demonstrated moderate performance with accuracies of 75.67%, 86.63%, and 88.94%, respectively. On the other hand, VGG-16 displayed a higher accuracy of 94.71%. Interestingly, the handcrafted features LBP, HOG, and BSIF outperformed all other models, achieving accuracies of 92.69%, 99%, and 99.13%, respectively. Similarly, the lightweight networks (DCTNet, DLNet, and MDFNet) achieved high accuracies of 99%.

In terms of recall, VGG-16 demonstrated reasonably good performance for detecting cyst, normal, stone, and tumor classes, with recall values of 0.88, 1, 0.97, and 0.95, respectively. The CCT model performed well in detecting cyst and stone class images, with recall values of 0.85 and 0.89, respectively. For the lightweight networks, DCTNet achieved perfect recall (1) for cyst and tumor classes, indicating its strong performance in detecting Gaussian images. Among the handcrafted features, LBP, HOG, and BSIF achieved high recall for all classes.

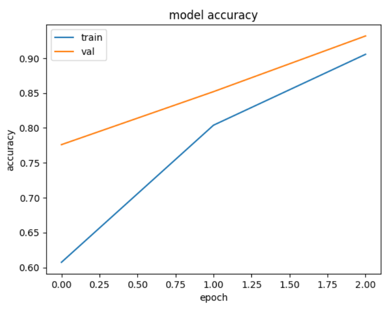
Considering precision, VGG-16 exhibited higher precision values (0.98, 0.96, 0.88, and 0.97) for the respective classes in the transfer-based approach. CCT stood out in the transformer-based approach with superior precision values (0.97, 0.87, 0.91, and 0.81). HOG demonstrated high precision values (0.98, 0.99, 0.99, and 0.99) in the handcrafted features-based approach. In the lightweight-based approach, all DCNet, DLNet, and MDFNet achieved high precision.

For the F1 score, VGG-16 achieved the highest scores (0.93, 0.98, 0.92, and 0.96) for cyst, normal, stone, and tumor classes, respectively. CCT demonstrated strong F1 scores (0.91, 0.88, 0.90, and 0.87) in the transformer-based approach. The lightweight-based approaches (DCTNet, DLNet, and MDFNet) and handcrafted features showed impressive F1 scores close to 1.

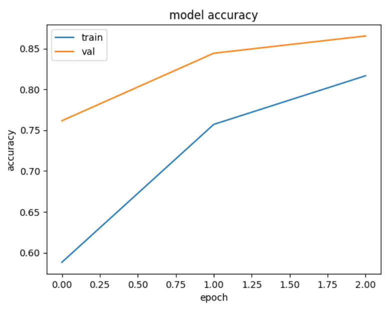
Table 3.5: The performance measures for all the models on the GAUSSIAN dataset

Models	Accuracy	Class	Precision	Recall(Sensitivity)	F1 Score	AUC
VGG16	94.71%	Cyst	0.98	0.88	0.93	0.993
		Normal	0.96	1	0.98	0.998
		Stone	0.88	0.97	0.92	0.991
		Tumor	0.97	0.95	0.96	0.995
Resnet	86.63%	Cyst	0.91	0.86	0.88	0.983
		Normal	0.85	0.91	0.88	0.980
		Stone	0.92	0.78	0.84	0.983
		Tumor	0.79	0.95	0.86	0.991
Inception v3	75.67%	Cyst	0.96	0.62	0.75	0.966
		Normal	0.82	0.86	0.84	0.969
		Stone	0.55	0.83	0.66	0.926
		Tumor	0.7	0.84	0.77	0.963
CCT	88.94%	Cyst	0.97	0.85	0.91	0.980
		Normal	0.87	0.89	0.88	0.975
		Stone	0.91	0.89	0.90	0.985
		Tumor	0.81	0.94	0.87	0.970
EANet	54.62%	Cyst	0.50	0.57	0.53	0.836
		Normal	O.83	0.53	0.65	0.882
		Stone	0.40	0.53	0.46	0.794
		Tumor	0.46	0.56	0.50	0.819
SWIN	65.29%	Cyst	0.58	0.71	0.64	0.891
		Normal	0.54	0.82	0.65	0.903
		Stone	0.63	0.63	0.63	0.875
		Tumor	0.85	0.56	0.68	0.906
LBP	92.69%	Cyst	0.87	0.87	0.87	0.915
		Normal	0.87	0.73	0.80	0.881
		Stone	0.73	0.84	0.78	0.841
		Tumor	0.82	0.86	0.84	0.887
HOG	99%	Cyst	0.98	0.99	0.98	0.988
		Normal	0.99	0.99	0.99	0.995
		Stone	0.99	0.98	0.99	0.992
		Tumor	0.99	1	0.99	0.995
BSIF	99.13%	Cyst	0.99	0.99	0.99	0.996
		Normal	0.98	0.99	0.99	0.989
		Stone	0.99	1	1	0.996
		Tumor	1	0.99	0.99	0.996
DCNet	99%	Cyst	1	1	1	0.998
		Normal	1	1		0.998
		Stone	1	0.99	1	0.998
		Tumor	1	1	1	1
DLNet	99%	Cyst	1	0.99	0.99	0.996
		Normal	0.98	1	0.99	0.998
		Stone	1	0.99	0.99	0.996
		Tumor	1	0.99	1	0.998
MDFNet	99%	Cyst	0.99	1	0.99	0.995
		Normal	0.99	1	0.99	0.994
		Stone	1	0.98	0.99	0.994
		Tumor	1	1	1	1

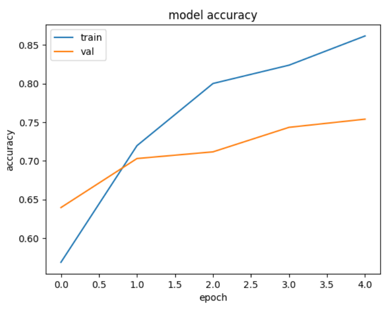
Figures 3.41, 3.42 summarizes the training and loss curves, demonstrating the improvement in classification performance of the CNN and Transformer models with increasing epochs.



Accuracy for VGG-16

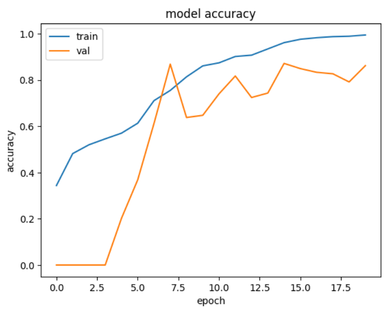


Accuracy for RESNet-50

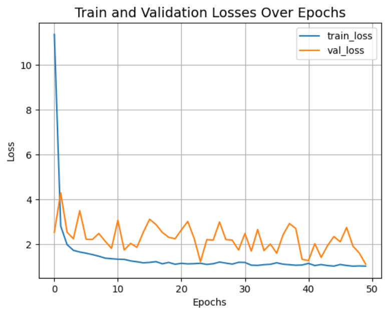


Accuracy for Inception-V3

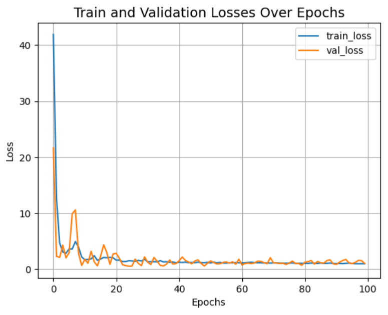
Figure 3.41: Accuracy curves for CNN Based Models Used On Gaussian Dataset.



Accuracy for CCT



Loss for EANet

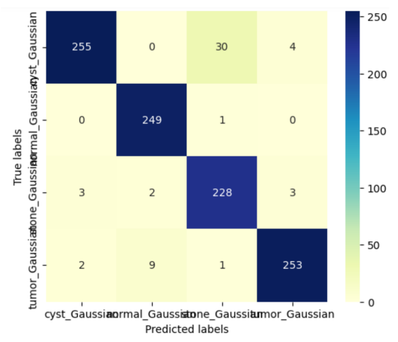


Loss for SWIN

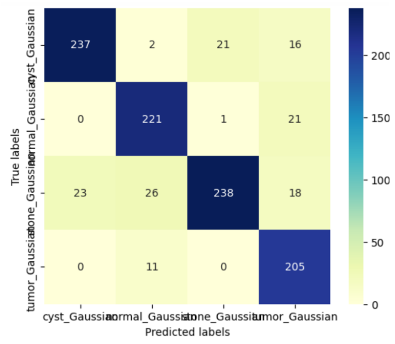
Figure 3.42: Accuracy and Loss curves for Transformer Based Models Used On Gaussian Dataset.

Figures 3.43, 3.44, 3.45, and 3.46 shows the normalized Confusion Matrices for Transfer and Transformer, Lightweight based models, and Handcrafted consecutively for gaussian dataset.

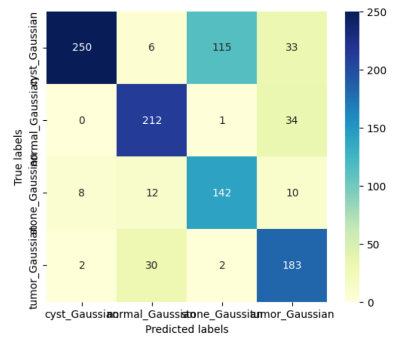
Figures 3.47, 3.48, 3.49, 3.50 provides the ROC curves for Transfer, Transformer, Lightweight based models, and Handcrafted consecutively for gaussian dataset.



Confusion Matrix for VGG-16

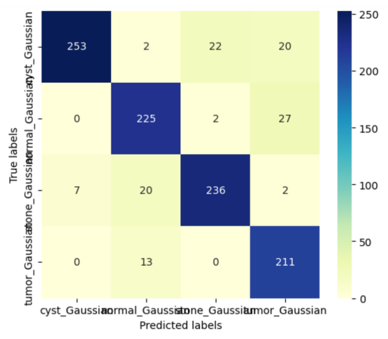


Confusion Matrix for RESNet-50

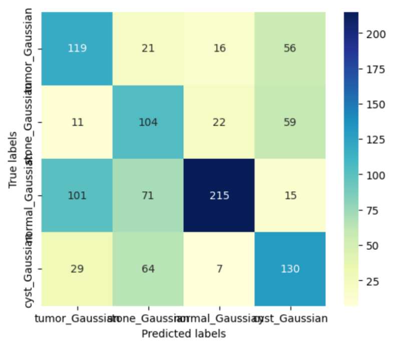


Confusion Matrix for Inception-V3

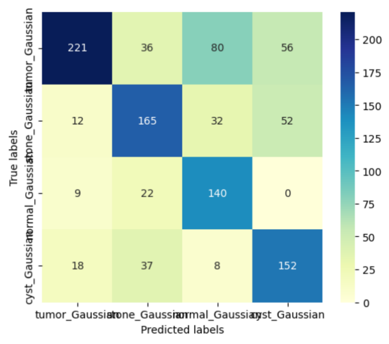
Figure 3.43: Confusion Matrix for CNN Based Models Used On Gaussian Dataset.



Confusion Matrix for CCT

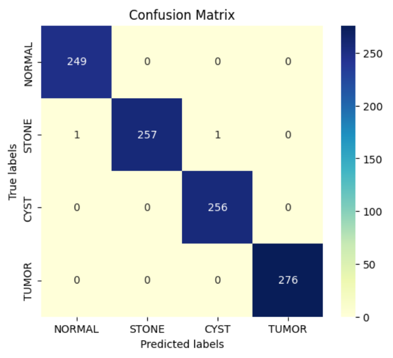


Confusion Matrix for EANet

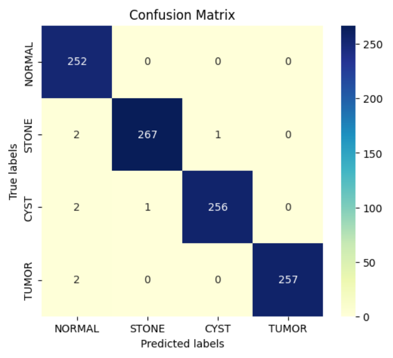


Confusion Matrix for SWIN

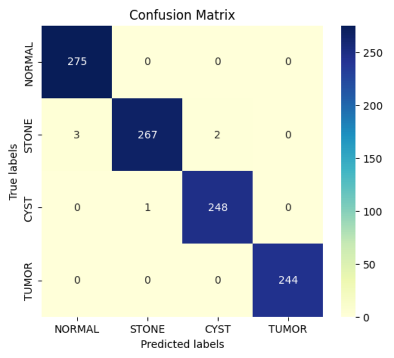
Figure 3.44: Confusion Matrix for Transformer Based Models Used On Gaussian Dataset.



Confusion Matrix for DCT-Net

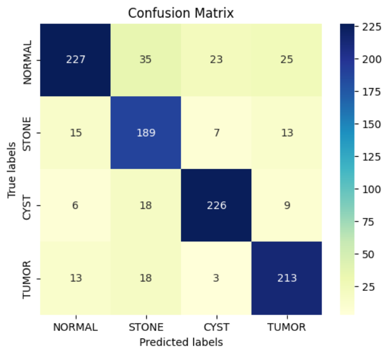


Confusion Matrix for DLNet

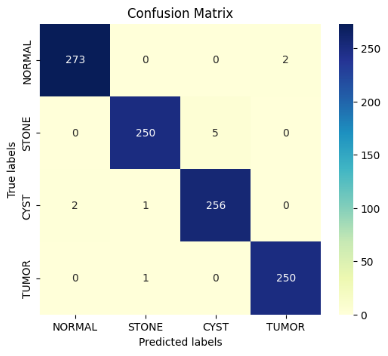


Confusion Matrix for MDFNet

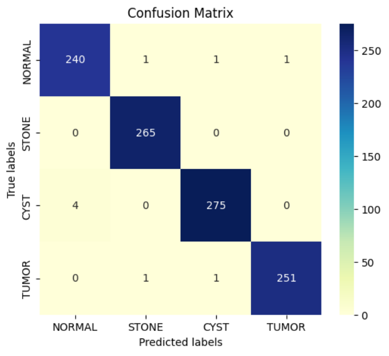
Figure 3.45: Confusion Matrix for Lightweight neural networks Based Models Used On Gaussian Dataset.



Confusion Matrix for LBP

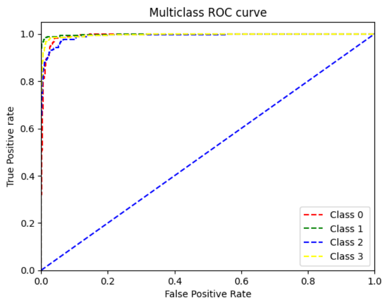


Confusion Matrix for HOG

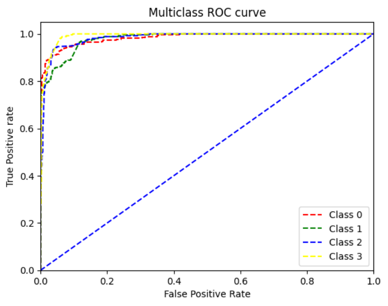


Confusion Matrix for BSIF

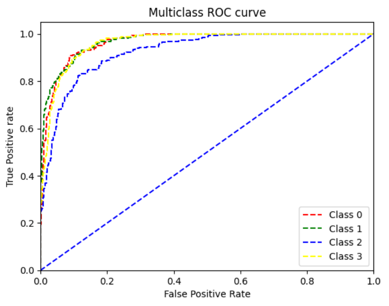
Figure 3.46: Confusion Matrix for Handcrafted features Based Models Used On Gaussian Dataset.



Roc Curve for VGG-16

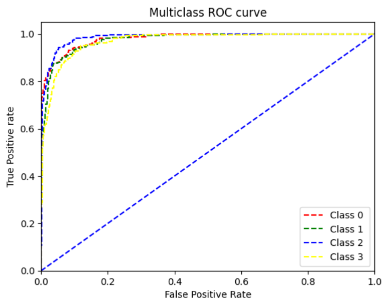


Roc Curve for RESNet-50

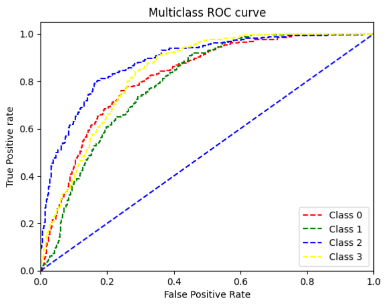


Roc Curve for Inception-v3

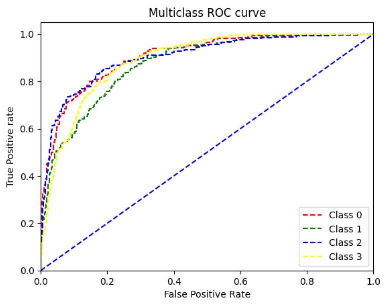
Figure 3.47: Roc Curves for CNN Based Models Used On Gaussian Dataset.



Roc Curve for CCT

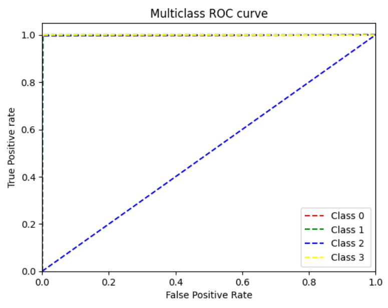


Roc Curve for EANet

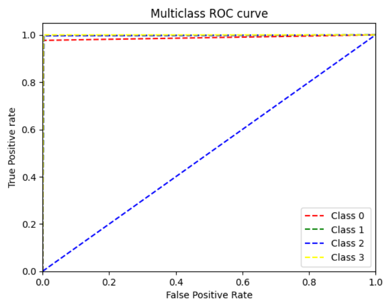


Roc Curve for SWIN

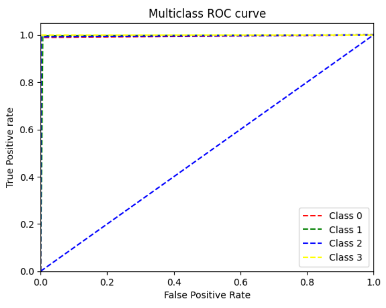
Figure 3.48: Roc Curves for Transformer Based Models Used On Gaussian Dataset.



Roc Curve for DCTNET

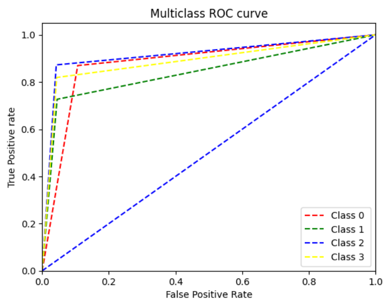


Roc Curve for DLNET

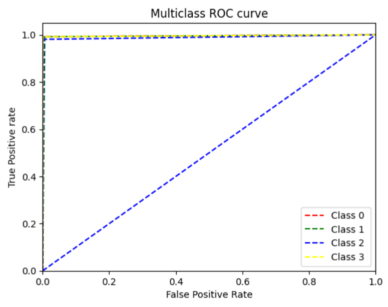


Roc Curve for MDFNET

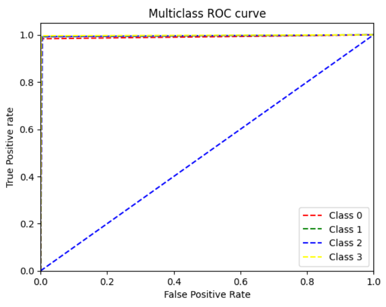
Figure 3.49: Roc Curves for Lightweight neural networks Based Models Used On Gaussian Dataset.



Roc Curve for LBP



Roc Curve for HOG



Roc Curve for BSIF

Figure 3.50: Roc Curves for Handcrafted features Based Models Used On Gaussian Dataset.

Experiment 6: performance using the Illumination Decrease dataset

According to the results presented in Table 3.6, our comparative analysis revealed varying performance among the different models with respect to our illumination decrease dataset. The Inception-v3 and EANet models exhibited lower performance, achieving accuracies of 83.65% and 80.58%, respectively. However, VGG-16, RESNet-50, CCT, and SWIN demonstrated high performance with accuracies of 96.44%, 90.29%, 97.69%, and 95.19%, respectively. Interestingly, the handcrafted features LBP, HOG, and BSIF outperformed all other models, achieving accuracies of 98.71%, 99%, and 98.84%, respectively. Similarly, the lightweight networks (DCTNet, DLNet, and MDFNet) achieved a high accuracy of 99%.

In terms of recall, VGG-16 demonstrated reasonably good performance for detecting cyst, normal, stone, and tumor classes, with recall values of 0.93, 0.95, 0.99, and 1, respectively. The CCT model performed well in detecting cyst and stone class images, with recall values of 0.99 and 0.98, respectively. For the lightweight networks, DCTNet achieved perfect recall (1) for cyst and tumor classes, indicating its strong performance in detecting images with decreased illumination. Among the handcrafted features, LBP, HOG, and BSIF achieved high recall for all classes.

Considering precision, VGG-16 exhibited higher precision values (1, 0.99, 0.93, and 0.94) for the respective classes in the transfer-based approach. CCT stood out in the transformer-based approach with superior precision values (0.96, 0.96, 0.98, and 1) compared to EANet and SWIN. HOG demonstrated precision values of 1 for all respective classes in the handcrafted features-based approach. In the lightweight-based approach, all DCNet, DLNet, and MDFNet achieved high precision.

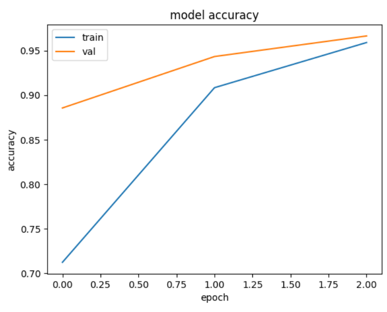
For the F1 score, VGG-16 achieved the highest scores (0.96, 0.97, 0.96, and 0.96) for cyst, normal, stone, and tumor classes, respectively. CCT demonstrated strong F1 scores (0.98, 0.97, 0.96, and 1) in the transformer-based approach. The lightweight-based approaches (DCTNet, DLNet, and MDFNet) and handcrafted features showed impressive F1 scores close to 1.

The same explanation applies as in the case of the illumination increase dataset

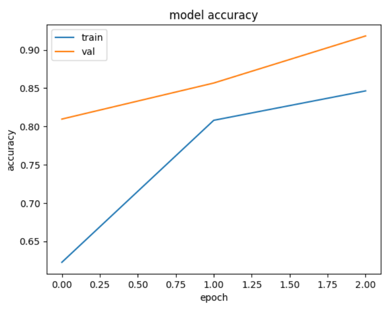
Table 3.6: The performance measures for all the models on the ILLUMINATION DECREASE dataset

Models	Accuracy	Class	Precision	Recall(Sensitivity)	F1 Score	AUC
VGG16	96.44%	Cyst	1	0.93	0.96	0.997
		Normal	0.99	0.95	0.97	0.997
		Stone	0.93	0.99	0.96	0.999
		Tumor	0.94	1	0.97	0.998
Resnet	90.29%	Cyst	0.97	0.81	0.88	0.991
		Normal	0.97	0.97	0.92	0.993
		Stone	0.85	0.92	0.89	0.988
		Tumor	0.92	0.93	0.93	0.984
Inception v3	83.65%	Cyst	0.90	0.82	0.86	0.979
		Normal	0.92	0.78	0.84	0.974
		Stone	0.75	0.90	0.82	0.971
		Tumor	0.78	0.88	0.83	0.972
CCT	97.69%	Cyst	0.96	0.99	0.98	0.996
		Normal	0.96	0.98	0.97	0.994
		Stone	0.98	0.94	0.96	0.993
		Tumor	1	1	1	1
EANet	80.58%	Cyst	0.38	0.98	0.54	0.965
		Normal	0.92	0.76	0.83	0.979
		Stone	0.92	0.66	0.77	0.955
		Tumor	1	1	1	1
SWIN	95.19%	Cyst	0.95	0.95	0.95	0.996
		Normal	0.93	0.94	0.94	0.995
		Stone	0.92	0.92	0.92	0.986
		Tumor	1	1	1	1
LBP	98.71%	Cyst	0.97	0.92	0.95	0.972
		Normal	0.97	0.97	0.97	0.979
		Stone	0.93	0.98	0.95	0.959
		Tumor	1	1	1	1
HOG	99%	Cyst	1	1	1	0.997
		Normal	1	1	1	0.997
		Stone	1	1	1	0.999
		Tumor	1	1	1	1
BSIF	98.84%	Cyst	0.99	0.99	0.99	0.995
		Normal	0.98	0.98	0.98	0.986
		Stone	0.98	0.98	0.98	0.987
		Tumor	1	1	1	1
DCNet	99.32%	Cyst	0.98	1	0.99	0.992
		Normal	0.99	0.99	0.99	0.992
		Stone	1	0.98	0.99	0.996
		Tumor	1	1	1	1
DLNet	99%	Cyst	0.96	1	0.98	0.979
		Normal	1	0.98	0.99	0.993
		Stone	1	0.98	0.99	0.997
		Tumor	1	1	1	0.999
MDFNet	99%	Cyst	0.98	1	0.99	0.992
		Normal	0.98	0.98	0.98	0.988
		Stone	1	0.99	0.99	0.998
		Tumor	1	1	1	0.997

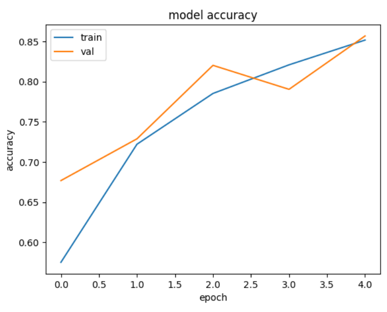
Figures 3.51, 3.52 summarizes the training and loss curves, demonstrating the improvement in classification performance of the CNN and Transformer models with increasing epochs.



Accuracy for VGG-16

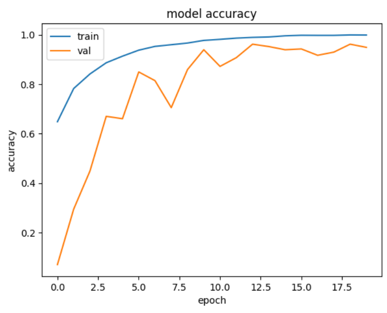


Accuracy for RESNet-50

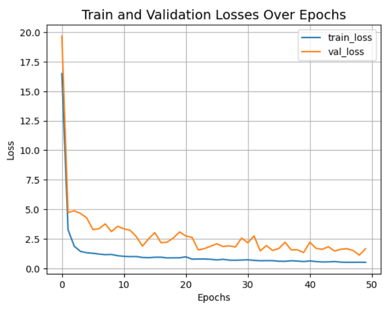


Accuracy for Inception-V3

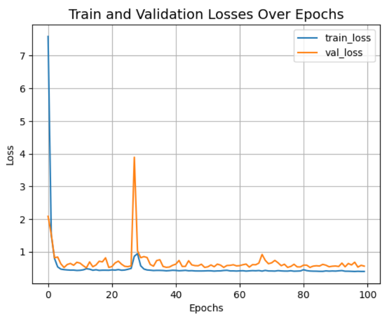
Figure 3.51: Accuracy curves for CNN Based Models Used On Illumination Decrease Dataset.



Accuracy for CCT



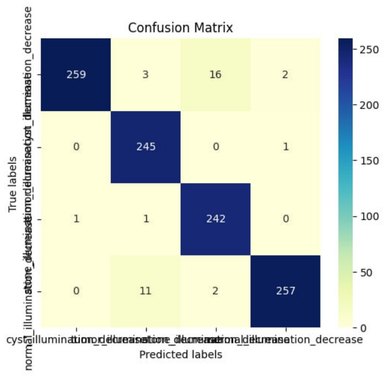
Loss for EANet



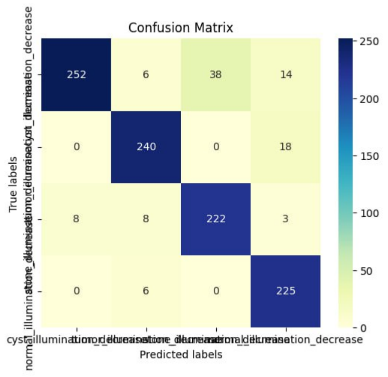
Loss for SWIN

Figure 3.52: Accuracy and Loss curves for Transformer Based Models Used On Illumination Decrease Dataset.

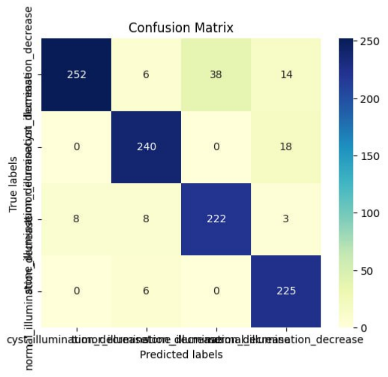
Figures 3.53, 3.54, 3.55, and 3.56 shows the normalized Confusion Matrices for Transfer and Transformer, Lightweight based models, and Handcrafted consecutively for illumination decrease dataset. Figures 3.57, 3.58, 3.59, 3.60 provides the ROC curves for Transfer, Transformer, Lightweight based models, and Handcrafted consecutively for illumination decrease dataset.



Confusion Matrix for VGG-16

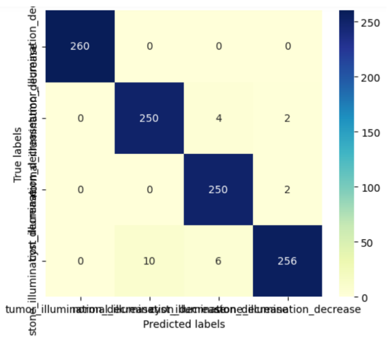


Confusion Matrix for RESNet-50

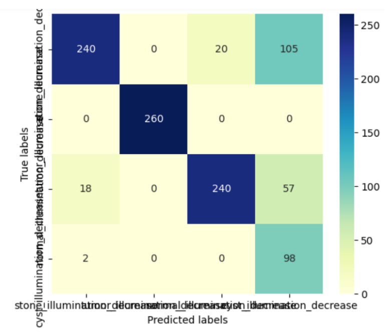


Confusion Matrix for Inception-V3

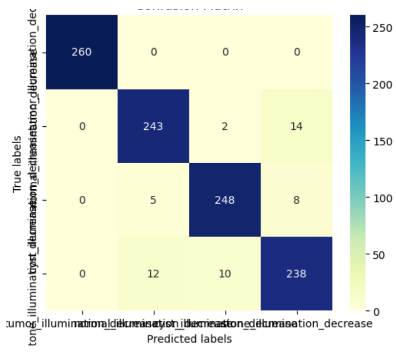
Figure 3.53: Confusion Matrix for CNN Based Models Used On Illumination Decrease Dataset.



Confusion Matrix for CCT

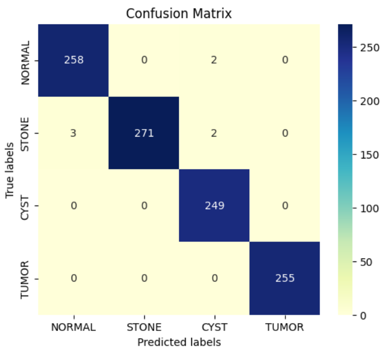


Confusion Matrix for EANet

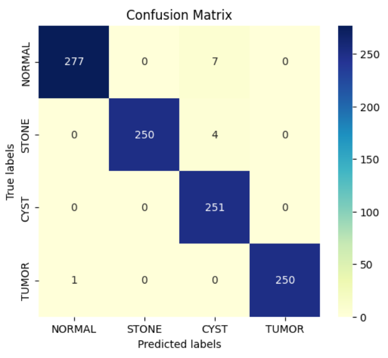


Confusion Matrix for SWIN

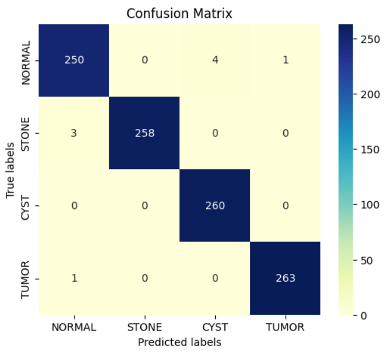
Figure 3.54: Confusion Matrix for Transformer Based Models Used On Illumination Decrease Dataset.



Confusion Matrix for DTC-Net

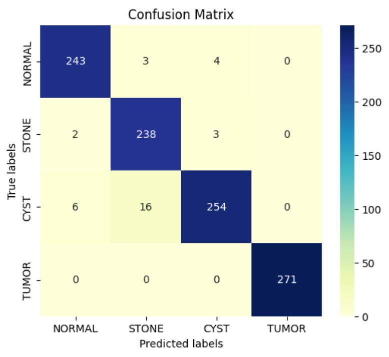


Confusion Matrix for DLNet

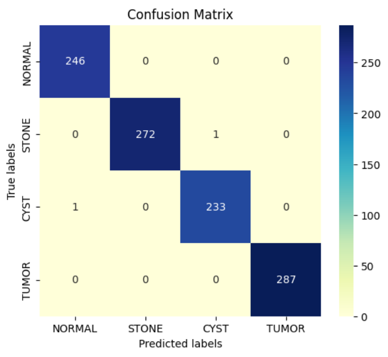


Confusion Matrix for MDFNet

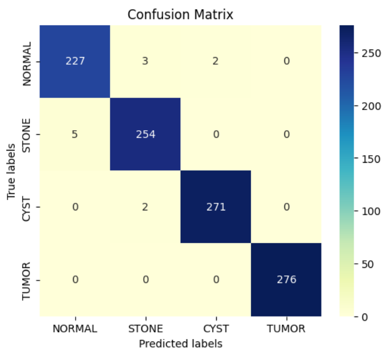
Figure 3.55: Confusion Matrix for Lightweight neural networks Based Models Used On Illumination Decrease Dataset.



Confusion Matrix for LBP

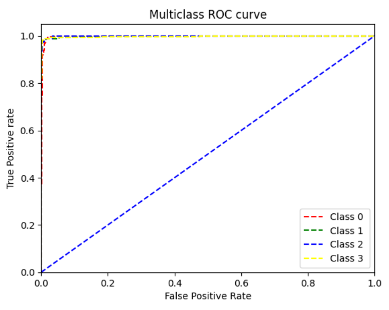


Confusion Matrix for HOG

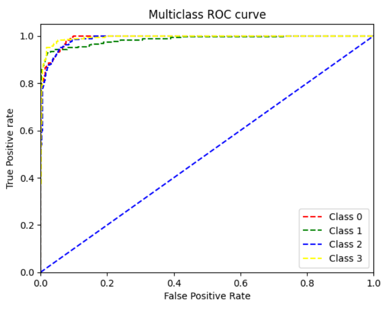


Confusion Matrix for BSIF

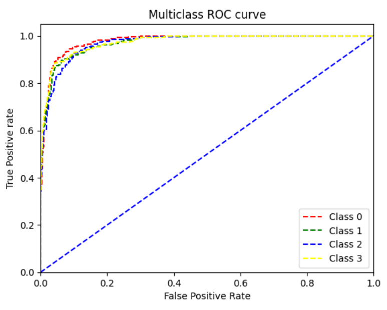
Figure 3.56: Confusion Matrix for Handcrafted features Based Models Used On Illumination Decrease Dataset.



Roc Curve for VGG-16

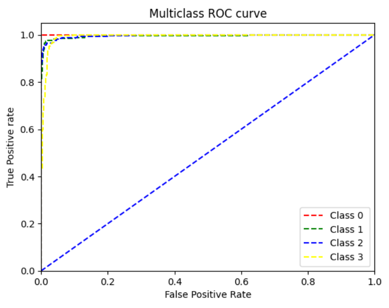


Roc Curve for RESNet-50

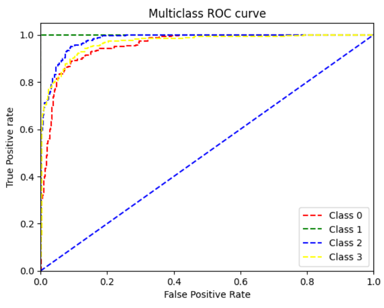


Roc Curve for Inception-v3

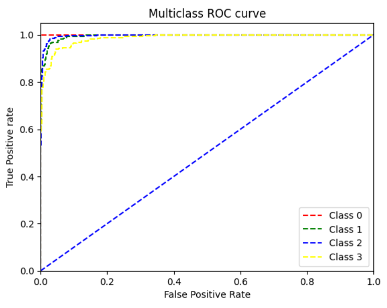
Figure 3.57: Roc Curves for CNN Based Models Used On Illumination Decrease Dataset.



Roc Curve for CCT

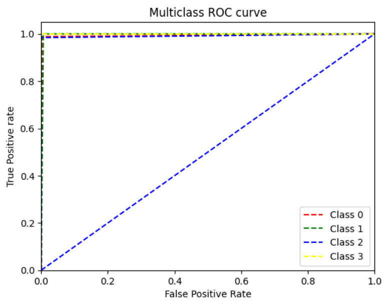


Roc Curve for EANet

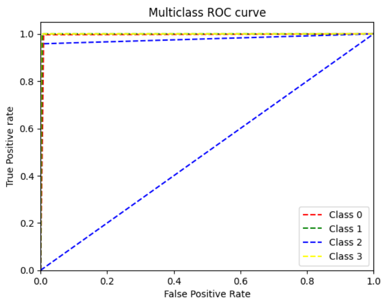


Roc Curve for SWIN

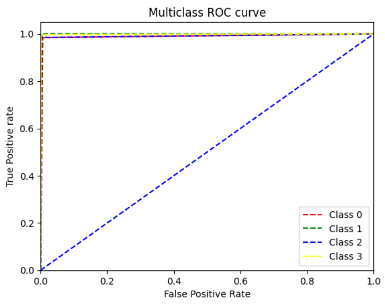
Figure 3.58: Roc Curves for Transformer Based Models Used On Illumination Decrease Dataset.



Roc Curve for DCTNET

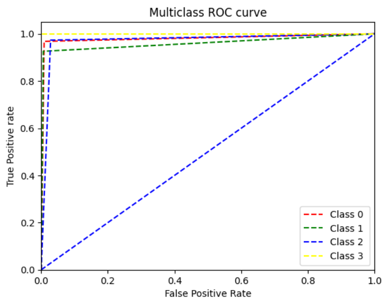


Roc Curve for DLNET

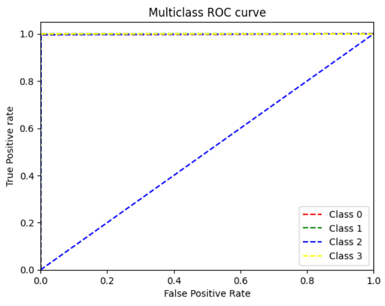


Roc Curve for MDFNET

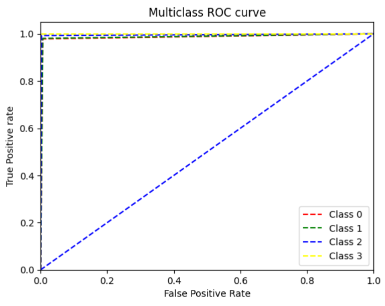
Figure 3.59: Roc Curves for Lightweight neural networks Based Models Used On Illumination Decrease Dataset.



Roc Curve for LBP



Roc Curve for HOG



Roc Curve for BSIF

Figure 3.60: Roc Curves for Handcrafted features Based Models Used On Illumination Decrease Dataset.

3.3 Conclusion

In this chapter, we have extensively showcased the outcomes of our analysis using various models, namely convolutional neural networks (CNN), handcrafted features, and lightweight neural networks. To assess the performance of each model, we employed multiple evaluation metrics, including accuracy and loss curves, confusion matrix, and ROC curves.

The accuracy and loss curves provide a visual representation of how well the models performed over iterations of training and validation. By analyzing these curves, we gain insights into the convergence and stability of the models, allowing us to make informed judgments about their efficacy.

Additionally, the confusion matrix helps us understand the classification performance of the models by providing a detailed breakdown of true positives, true negatives, false positives, and false negatives.

Furthermore, the ROC (Receiver Operating Characteristic) curves allow us to analyze the trade-off between true positive rate and false positive rate for various classification thresholds.

Throughout the chapter, we thoroughly discussed the results obtained from each model, considering their strengths, weaknesses, and implications in the context of kidney disease analysis.

Conclusion

In this research, we utilized the "CT KIDNEY DATASET: Normal-Cyst-Tumor and Stone" dataset. To enhance our analysis, we applied modifications to the original images, resulting in new datasets that incorporated various transformations. These transformations included blurring, Gaussian noise, changes in illumination (both increase and decrease), and occlusion.

Our study involved the development of twelve models, each serving a specific purpose. Among these models, three were based on recent state-of-the-art variants of Vision transformers (EANet, CCT, and Swin transformers), three were CNN models (Resnet, Vgg16, and Inception v3), three employed handcrafted features (LBP, HOG, and BSIF), and three were lightweight neural networks (DLNet, MDFNet, and DCTNet).

Upon comparing the performance of these models, we observed interesting outcomes. In the original dataset, the BSIF handcrafted features exhibited superior accuracy, achieving an impressive accuracy of 99.51%. In the BLUR dataset, the lightweight neural networks outperformed all other models, achieving an accuracy of 99%.

However, in the ILLUMINATION INCREASE dataset, we noted a decline in classification results when using Vision transformers models (EANet and SWIN). Conversely, in the OCCLUSION dataset, the lightweight neural networks surpassed all other models in terms of accuracy. In the GAUSSIAN dataset, the lightweight neural networks once again outperformed all other models, achieving an accuracy of 99%. Similarly, in the ILLUMINATION DECREASE dataset, the lightweight neural networks demonstrated superior accuracy.

Considering the outstanding accuracy achieved by our model, we believe it can play a vital role in the detection of kidney tumors, cysts, and stones. This has the potential to alleviate the pain and suffering experienced by patients.

Overall, our research highlights the effectiveness of different models and transformations in the analysis of kidney diseases, providing valuable insights for future advancements in the field.

Table 3.7: CNN Accuracy.

Model	VGG-16	RESNet-50	Inception-v3
Accuracy	87.50%	90.29%	82.60%

Table 3.8: Transformers Accuracy.

Model	CCT	EANet	SWIN
Accuracy	88.65%	52.5%	47.88%

Table 3.9: Handcrafted Accuracy.

Model	LBP	HOG	BSIF
Accuracy	98.26%	99%	99.58%

Table 3.10: Lightweight Accuracy.

Model	DLNet	DCTNet	MDFNet
Accuracy	99%	99%	99%

Bibliography

[1] Chris M Bishop. Neural networks and their applications. *Review of scientific instruments*, 65(6):1803–1832, 1994.

[2] Syed Muhammad Anwar, Muhammad Majid, Adnan Qayyum, Muhammad Awais, Majdi Alnowami, and Muhammad Khurram Khan. Medical image analysis using convolutional neural networks: a review. *Journal of medical systems*, 42:1–13, 2018.

[3] K. Demaagd, A. Oliver, N. Oostendorp, and K. Scott. *Practical Computer Vision with SimpleCV*. Oreilly and Associate Series. O'Reilly Media, Incorporated, 2012.

[4] Hussam Qassim, Abhishek Verma, and David Feinzimer. Compressed residual-vgg16 cnn model for big data places image recognition. In *2018 IEEE 8th annual computing and communication workshop and conference (CCWC)*, pages 169–175. IEEE, 2018.

[5] Bishwas Mandal, Adaeze Okeukwu, and Yihong Theis. Masked face recognition using resnet-50. *arXiv preprint arXiv:2104.08997*, 2021.

[6] M.A. Wani, M. Kantardzic, and M. Sayed-Mouchaweh. *Deep Learning Applications*. Advances in Intelligent Systems and Computing. Springer Nature Singapore, 2020.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[8] Meng-Hao Guo, Zheng-Ning Liu, Tai-Jiang Mu, and Shi-Min Hu. Beyond self-attention: External attention using two linear layers for visual tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[9] Md Nazmul Islam, Mehedi Hasan, Md Kabir Hossain, Md Golam Rabiul Alam, Md Zia Uddin, and Ahmet Soylu. Vision transformer and explainable transfer learning models for auto detection of kidney cyst, stone and tumor from ct-radiography. *Scientific Reports*, 12(1):11440, 2022.

[10] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[11] Kaplan Kaplan, Yılmaz Kaya, Melih Kuncan, and H Metin Ertunç. Brain tumor classification using modified local binary patterns (lbp) feature extraction methods. *Medical hypotheses*, 139:109696, 2020.

[12] Pei-Yin Chen, Chien-Chuan Huang, Chih-Yuan Lien, and Yu-Hsien Tsai. An efficient hardware implementation of hog feature extraction for human detection. *IEEE Transactions on Intelligent Transportation Systems*, 15(2):656–662, 2013.

[13] Oussama Aiadi and Belal Khaldi. A fast lightweight network for the discrimination of covid-19 and pulmonary diseases. *Biomedical Signal Processing and Control*, 78:103925, 2022.

[14] Cong Jie Ng and Andrew Beng Jin Teoh. Dctnet: A simple learning-free approach for face recognition. In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 761–768. IEEE, 2015.

[15] Işıl AKSAKALLI, Sibel KAÇDIOĞLU, and Y Sinan HANAY. Kidney x-ray images classification using machine learning and deep learning methods. *Balkan Journal of Electrical and Computer Engineering*, 9(2):144–151, 2021.

[16] S Sudharson and Priyanka Kokil. An ensemble of deep neural networks for kidney ultrasound image classification. *Computer Methods and Programs in Biomedicine*, 197:105709, 2020.

[17] Xu Fu, Huaiqin Liu, Xiaowang Bi, Xiao Gong, et al. Deep-learning-based ct imaging in the quantitative evaluation of chronic kidney diseases. *Journal of Healthcare Engineering*, 2021, 2021.

[18] Qiang Zheng, Susan L Furth, Gregory E Tasian, and Yong Fan. Computer-aided diagnosis of congenital abnormalities of the kidney and urinary tract in children based on ultrasound imaging data by integrating texture image features and deep transfer learning image features. *Journal of pediatric urology*, 15(1):75–e1, 2019.

[19] Anushri Parakh, Hyunkwang Lee, Jeong Hyun Lee, Brian H Eisner, Dushyant V Sahani, and Synho Do. Urinary stone detection on ct images using deep convolutional neural networks: evaluation of model performance and generalization. *Radiology: Artificial Intelligence*, 1(4):e180066, 2019.

[20] Kadir Yildirim, Pinar Gundogan Bozdag, Muhammed Talo, Ozal Yildirim, Murat Karabatak, and U Rajendra Acharya. Deep learning model for automated kidney stone detection using coronal ct images. *Computers in biology and medicine*, 135:104569, 2021.

[21] Md Humaion Kabir Mehedi, Ehteshamul Haque, Sameen Yasir Radin, and Md Abrar Ur Rahman. *Segmentation based Kidney Tumor Classification using Deep Neural Network*. PhD thesis, Brac University, 2022.

[22] Liam James Glennie England. *Renal Calculi Detection Using Convolutional Neural Networks*. PhD thesis, Universitat Politècnica de València, 2022.

[23] Ahmet Çınar, Muhammed Yıldırım, and Yeşim Eroğlu. Classification of pneumonia cell images using improved resnet50 model. *Traitement du Signal*, 38(1):165–173, 2021.

[24] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133, 1943.

[25] Varun Gulshan, Lily Peng, Marc Coram, Martin C Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Jama*, 316(22):2402–2410, 2016.

[26] Muhammad Imran Razzak, Saeeda Naz, and Ahmad Zaib. Deep learning for medical image processing: Overview, challenges and the future. *Classification in BioApps: Automation of Decision Making*, pages 323–350, 2018.

[27] Praba Hridayami, I Ketut Gede Darma Putra, and Kadek Suar Wibawa. Fish species recognition using vgg16 deep convolutional neural network. *Journal of Computing Science and Engineering*, 13(3):124–130, 2019.

[28] Haoyan Yang, Jiangong Ni, Jiyue Gao, Zhongzhi Han, and Tao Luan. A novel method for peanut variety identification and classification by improved vgg16. *Scientific Reports*, 11(1):1–17, 2021.

[29] Ryoji Kadota, Hiroki Sugano, Masayuki Hiromoto, Hiroyuki Ochi, Ryusuke Miyamoto, and Yukihiro Nakamura. Hardware architecture for hog feature extraction. In *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 1330–1333. IEEE, 2009.

[30] Bilal Attallah, Amina Serir, Youssef Chahir, and Abdelwahhab Boudjelal. Histogram of gradient and binarized statistical image features of wavelet subband-based palmprint features extraction. *Journal of Electronic Imaging*, 26(6):063006–063006, 2017.

[31] Maulisa Oktiana, Fitri Arnia, Yuwaldi Away, and Khairul Munadi. Features for cross spectral image matching: A survey. *Bulletin of Electrical Engineering and Informatics*, 7(4):552–560, 2018.

[32] Iqbal H Sarker. Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, 2(6):420, 2021.

[33] Yi-Lun Liao, Sertac Karaman, and Vivienne Sze. Searching for efficient multi-stage vision transformers. *arXiv preprint arXiv:2109.00642*, 2021.

[34] Yifan Xu, Huapeng Wei, Minxuan Lin, Yingying Deng, Kekai Sheng, Mengdan Zhang, Fan Tang, Weiming Dong, Feiyue Huang, and Changsheng Xu. Transformers in computational visual media: A survey. *Computational Visual Media*, 8:33–62, 2022.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[36] Edoardo Debenedetti and Carmela Troncoso—EPFL. *Adversarially robust vision transformers*. PhD thesis, Master's thesis, Swiss Federal Institue of Technology, Lausanne (EPFL), 2022.

[37] Aqeel Iftikhar Jajja, Assad Abbas, Hasan Ali Khattak, Gniewko Niedbała, Abbas Khalid, Hafiz Tayyab Rauf, and Sebastian Kujawa. Compact convolutional transformer (cct)-based approach for whitefly attack detection in cotton crops. *Agriculture*, 12(10):1529, 2022.

[38] Emily R Bartusiak and Edward J Delp. Synthesized speech detection using convolutional transformer-based spectrogram analysis. In *2021 55th Asilomar Conference on Signals, Systems, and Computers*, pages 1426–1430. IEEE, 2021.

[39] Inam Ullah Khan, Mohaimenul Azam Khan Raiaan, Kaniz Fatema, Sami Azam, Rafi ur Rashid, Saddam Hossain Mukta, Mirjam Jonkman, and Friso De Boer. A computer-aided diagnostic system to identify diabetic retinopathy, utilizing a modified compact convolutional transformer and low-resolution images to reduce computation time. *Biomedicines*, 11(6):1566, 2023.

[40] Sadaf Khademi, Shahin Heidarian, Parnian Afshar, Farnoosh Naderkhani, Anastasia Oikonomou, Konstantinos N Plataniotis, and Arash Mohammadi. Spatio-temporal hybrid fusion of cae and swin transformers for lung cancer malignancy prediction. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

Kidney cysts, tumors, and stones pose significant health risks to individuals, potentially leading to kidney failure if not effectively managed. However, the shortage of nephrologists and kidney specialists globally presents a significant challenge in providing timely and accurate diagnoses.

In this research, we aim to address this challenge by leveraging artificial intelligence (AI) techniques for the detection of kidney cysts, tumors, stones, and normal conditions this was based on early research that made a comparative study between cnn models and transformer variants. To expand that research Our study encompasses an invistigation involving diverse models, including CNN models, transformer variants, lightweight neural networks, and handcrafted features. We evaluate these models on a dataset of 12,446 unique images extracted from abdominal CT scans. To enhance the comprehensiveness of our investigation, we introduce several modifications to the original dataset, incorporating Gaussian noise, blur, illumination variations (both increase and decrease), and occlusion. These modifications aim to simulate real-world conditions and imaging artifacts, providing a more realistic and challenging dataset for evaluation purposes.

Upon comparing the performance of these models, we observed interesting outcomes. In the original dataset, the BSIF handcrafted features exhibited superior accuracy, achieving an impressive accuracy of 99.51%. In the BLUR dataset, the lightweight neural networks outperformed all other models, achieving an accuracy of 99%. However, in the ILLUMINATION INCREASE dataset, we noted a decline in classifi-cation results when using Vision transformers models (EANet and SWIN). Conversely, in the OCCLUSION dataset, the lightweight neural networks surpassed all other models in terms of accuracy. In the GAUSSIAN dataset, the lightweight neural networks once again outperformed all other models, achieving an accuracy of 99%. Similarly, in the ILLUMI- NATION DECREASE dataset, the lightweight neural networks demonstrated superior accuracy.
key Words:
Deep Learning, Computer Vision, CNN(Convolutional Neural Network), Vision Transformers, Handcrafted features, Lightweight neural networks

Les kystes, les tumeurs et les calculs rénaux présentent des risques importants pour la santé des individus, pouvant entraîner une insuffisance rénale s'ils ne sont pas diagnostiqués et gérés efficacement. Cependant, la pénurie de néphrologues et de spécialistes des reins à l'échelle mondiale constitue un défi majeur pour des diagnostics précis et rapides.

Dans le cadre de cette recherche, nous visons à relever ce défi en exploitant des techniques d'intelligence artificielle (IA) pour la détection des kystes rénaux, des tumeurs, des calculs et des conditions normales. Cette étude s'appuie sur des recherches antérieures ayant réalisé une étude comparative entre les modèles CNN et les variantes des transformateurs. Pour étendre cette recherche, notre étude englobe une investigation portant sur divers modèles, notamment des modèles CNN, des variantes de transformateurs, des réseaux neuronaux légers et des caractéristiques artisanales. Nous évaluons ces modèles sur un ensemble de données comprenant 12 446 images uniques extraites de scanners abdominaux CT. Afin d'améliorer la qualité et l'exhaustivité de notre investigation, nous introduisons plusieurs modifications dans l'ensemble de données d'origine, en incorporant du bruit gaussien, du flou, des variations d'éclairage (augmentation et diminution) et de l'occlusion. Ces modifications visent à simuler des conditions réelles et des artefacts d'imagerie, fournissant ainsi un ensemble de données plus réaliste et plus difficile à évaluer.

En comparant les performances de ces modèles, nous avons observé des résultats intéressants. Dans l'ensemble de données d'origine, les caractéristiques artisanales BSIF

ont affiché une précision supérieure, atteignant une impressionnante précision de 99,51 %. Dans l'ensemble de données flou, les réseaux neuronaux légers ont surpassé tous les autres modèles, atteignant une précision de 99 % Cependant, dans l'ensemble de données d'augmentation d'éclairage, nous avons constaté une baisse des résultats de classification lors de l'utilisation des modèles de transformateurs de vision (EANet et SWIN). En revanche, dans l'ensemble de données d'occlusion, les réseaux neuronaux légers ont surpassé tous les autres modèles en termes de
Mots-clés:
Apprentissage profond, Vision par ordinateur, CNN (Réseau de neurones convolutifs), Transformers de vision, Caractéristiques artisanales, Réseaux neuronaux légers.

تشكل الكيسات والأورام والحصى الكلوية مخاطر صحية هامة للأفراد، وقد تؤدي في حال عدم التعامل معها بشكل فعال إلى الفشل الكلوي. ومع ذلك، يشكل نقص أخصائيي الكلى وأطباء الكلوة في جميع أنحاء العالم تحديًا كبيرًا في تقديم تشخيصات سريعة ودقيقة.

في هذا البحث، نهدف إلى معالجة هذا التحدي من خلال استغلال تقنيات الذكاء الاصطناعي (AI) للكشف عن الكيسات الكلوية والأورام والحصى والحالات الطبيعية، وقد اعتمد هذا البحث على أبحاث سابقة قامت بدراسة مقارنة بين نماذج الشبكات العصبية المتكررة والنماذج المتحولة. لتوسيع هذا البحث، تضمنت دراستنا تحقيقًا يشمل نماذج متنوعة، بما في ذلك نماذج الشبكات العصبية المتكررة والنماذج المتحولة وشبكات العصب الخفيفة والميزات المصنوعة يدويًا. قمنا بتقييم هذه النماذج باستخدام مجموعة بيانات تحتوي على ١٢،٤٤٦ صورة فريدة تم استخراجها من فحوصات (CT) البطنية. ولتعزيز شمولية تحقيقنا، قمنا بإدخال عدة تعديلات على مجموعة البيانات الأصلية، بما في ذلك إضافة ضوضاء غاوسية وتشويش وتغييرات في الإضاءة (زيادة وانخفاض) والحجب. تهدف هذه التعديلات إلى محاكاة ظروف واقعية وفنون تصويرية، وتوفير مجموعة بيانات أكثر واقعية وتحديًا لأغراض التقييم.

عند مقارنة أداء هذه النماذج، لاحظنا نتائج مثيرة للاهتمام. في مجموعة البيانات الأصلية، أظهرت الميزات المصنوعة يدويًا من نوع (BSIF) دقة متفوقة، حيث حققت دقة مبهرة تبلغ ٩٩،٥١% . في مجموعة البيانات المشوشة، تفوقت شبكات العصب الخفيفة على جميع النماذج الأخرى، حيث حققت دقة تبلغ ٩٩%. ومع ذلك، في مجموعة البيانات المتعلقة بزيادة الإضاءة، لاحظنا انخفاضًا في نتائج التصنيف عند استخدام نماذج تحول الرؤية ((EANet SWIN)) . وعلى العكس من ذلك، في مجموعة البيانات المتعلقة بالتعتيم ، تفوقت شبكات العصب الخفيفة على جميع النماذج الأخرى من حيث الدقة. في مجموعة البيانات المتعلقة بالضوضاء الغاوسية، قامت شبكات العصب الخفيفة مرة أخرى بتفوق على جميع النماذج الأخرى، حيث حققت دقة تبلغ ٩٩%. بالمثل، في مجموعة البيانات المتعلقة بانخفاض الإضاءة، أظهرت شبكات العصب الخفيفة دقة متفوقة.
الكلمات الرئيسية:
التعلم العميق، رؤية الحاسوب، شبكة عصبية تحويلية ((CNN)) ، التحويلات البصرية، الميزات

الحرفية، شبكات عصبية خفيفة الوزن.