People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University Of Kasdi Merbah Ouargla

Faculty of Information Technology and Communication

Department Of Information Technology

Domain: Mathematics and Computer Sciences

Field: Information Technology

Specialty: Fundamental Computer Sciences

Presented by: Mahcene Marwa and Bettayeb Hadjer

THEME

Brain Tumor Detection Using Deep Learning

The jury:

Dr.Korichi Maarouf	**President**	**UKM OUARGLA**
Dr. AIADI Ouassama	**Supervisor**	**UKM OUARGLA**
Dr.Benbezzian Mohamed	**Examiner**	**UKM OUARGLA**

AcademicYear : 2022/2023

Abstract

Brain tumor is an abnormal growth of cells inside the brain. It can arise from the supportive nerve tissues in the brain. Brain tumors are typically diagnosed using magnetic resonance imaging (MRI) and X-ray imaging. The symptoms of a brain tumor vary and depend on its size and location. Common symptoms include persistent and severe headaches, nausea, seizures, and more. The outcomes of brain tumors vary depending on the type of tumor and its stage. Research on the treatment of brain tumors is still ongoing.

In this thesis, we are interested in developing models capable of detecting brain tumors from MRI images using deep learning. We evaluate the individual performance of each of convolutional neural network (CNN) and convolutional auto-encoder (CAE) along with Support vector Machine (SVM) as a classifier. In particular, we opt for using the well-known pre-trained network namely VGG-16, ResNet-50 and the DenseNet-121.

To further improve the obtained results, we consider using different fusion schemes to combine the results of the different deep architectures. To validate the proposed approach, we conduct experiments on a public dataset of brain tumors that is made up of

(4600) images, we get 97.79% accuracy as a result of convolutional neural network (CNN), and 93% result of convolutional auto-encoder, ect.

Keywords: Brain-tumor, Deep Learning, CNN, convolutional auto-encoder (CAE), Medical Imaging, fusion, MRI.

Résumé :

Les tumeurs cérébrales sont des croissances anormales de cellules à l'intérieur du cerveau. Elles peuvent provenir des tissus nerveux de soutien dans le cerveau. Les tumeurs cérébrales sont généralement diagnostiquées à l'aide de l'imagerie par résonance magnétique (IRM) et de l'imagerie par rayons X. Les symptômes d'une tumeur cérébrale varient en fonction de sa taille et de son emplacement. Les symptômes courants comprennent des maux de tête persistants et sévères, des nausées, des convulsions, et d'autres encore. Les résultats des tumeurs cérébrales varient en fonction du type de tumeur et de son stade. La recherche sur le traitement des tumeurs cérébrales est toujours en cours.

Dans cette thèse, nous nous intéressons à développer des modèles capables de détecter les tumeurs cérébrales à partir d'images IRM en utilisant l'apprentissage profond. Nous évaluons les performances individuelles de chaque réseau de neurones convolutionnel (CNN) et d'auto-encodeur convolutionnel (CAE), ainsi que la machine à vecteurs de support (SVM) en tant que classifieur. En particulier, nous utilisons les réseaux pré-entraînés bien connus, à savoir VGG-16, ResNet-50 et DenseNet-121.

Pour améliorer davantage les résultats obtenus, nous envisageons d'utiliser différentes techniques de fusion pour combiner les résultats des différentes architectures d'apprentissage profond. Pour valider l'approche proposée, nous menons des expériences sur un ensemble de données public composé de 4 600 images de tumeurs cérébrales. Nos résultats démontrent une précision de 97,79% pour le CNN et une précision de 93% pour le CAE.

Mots-clés : Tumeur cérébrale, Apprentissage profond, CNN, Auto-encodeur convolutionnel (CAE), Imagerie médicale, Fusion, IRM.

ملخص

ورم الدماغ هو نمو غير طبيعي للخلايا داخل الدماغ. يمكن أن تنشأ من الأنسجة العصبية الداعمة في الدماغ. يتم تشخيص أورام الدماغ عادةً باستخدام التصوير بالرنين المغناطيسي (MRI) والتصوير بالأشعة السينية. تختلف أعراض ورم المخ وتعتمد على حجمه وموقعه. تشمل الأعراض الشائعة الصداع المستمر والشديد والغثيان والنوبات وغيرها. تختلف نتائج أورام المخ باختلاف نوع الورم ومرحلته. لا تزال الأبحاث حول علاج أورام المخ مستمرة.

في هذه الأطروحة ، نحن مهتمون بتطوير نماذج قادرة على اكتشاف أورام الدماغ من صور التصوير بالرنين المغناطيسي باستخدام التعلم العميق.نقوم بتقييم الأداء الفردي لكل من الشبكات العصبية التلافيفية (CNN) والتشفير التلقائي التلافيفي (CAE) جنبًا إلى جنب مع Support Vector Machine (SVM) كمصنف. على وجه الخصوص ، نختار استخدام الشبكة المعروفة مسبقًا وهي VGG-16 و ResNet-50 و DenseNet-121.

لزيادة تحسين النتائج التي تم الحصول عليها ، نعتبر استخدام مخططات اندماج مختلفة لدمج نتائج البنى العميقة المختلفة. للتحقق من صحة النهج المقترح ، نجري تجارب على مجموعة بيانات عامة لأورام الدماغ المكونة من (4600) صورة ، حصلنا على دقة 97.79٪ نتيجة للشبكة العصبية التلافيفية (CNN) ، و 93٪ نتيجة للتشفير التلقائي التلافيفي ، إلخ.

الكلمات المفتاحية: ورم الدماغ ، التعلم العميق ، الشبكات العصبية التلافيفية(CNN)،التشفير التلقائي التلافيفي(CAE)، الصور الطبية،الاندماج، MRI.

List of Contents

List of Figures

List of Tables

Dedication

First and foremost, I express my deepest gratitude to God Almighty for His immense blessings and for guiding me to this pivotal moment in my life. It is by His grace that I stand here today.

I dedicate this momentous achievement to my beloved mother and father, as they deserve the greatest credit, second only to God, for their unwavering support, prayers, and countless sacrifices. They have been my pillars of strength throughout this journey.

To my brothers, Nihad and Ayoub, Iman, and firas I am indebted to you for your constant encouragement and belief in my abilities. Your presence has been a source of inspiration and motivation for me.

I would also like to express my gratitude to the entire extended family, with a special mention to grandfather, who was a guiding light in my life. May his soul rest in peace.

A heartfelt appreciation goes to our esteemed professor, Dr.AIADIOuassama for his invaluable guidance and mentorship throughout the entire duration of my academic pursuits. His wisdom and expertise have shaped my intellectual growth in immeasurable ways.

Furthermore, I extend my thanks to Ms. AmiratHanan for her unwavering assistance . I am also grateful to the members of jury for accepting the assessment of our research.

Last but not least, I would like to express my deepest gratitude to my dear friend Hadjer. Your unwavering friendship and collaboration during our thesis work and throughout our university journey have been invaluable.

MahceneMarwa.

Dedication

First and foremost, I express my deepest gratitude to God Almighty for His immense blessings and for guiding me to this pivotal moment in my life. It is by His grace that I stand here today.

I would like to express my deepest gratitude to my supervisor Dr.aiadioussama for their invaluable guidance, support, and expertise throughout this project. Their unwavering commitment and encouragement have been instrumental in shaping my research journey and pushing me to excel.

My heartfelt appreciation goes to my family and friends for their unconditional love, understanding, and unwavering belief in my abilities. Their constant encouragement and support have been the pillars of my strength.

I am also grateful to my colleagues and peers for their stimulating discussions, collaboration, and constructive feedback that have contributed to the development and refinement of my work.

I would like to acknowledge the University Of Kasdi Merbah Ouargla for providing the necessary resources and infrastructure, enabling me to carry out this research.

Finally, I am indebted to all the participants and individuals who generously contributed their time and expertise, without whom this research would not have been possible.

Thank you to everyone who has played a part, big or small, in making this project a reality. Your contributions have left an indelible mark on my personal and professional growth.

Bettayebhadjer.

Gneral introduction

I.1Introduction

Brain tumors are a serious medical condition that has been greatly impacted by advancements in technology, particularly in the field of radiology. The use of radiology, including ultrasound, has revolutionized medical diagnostics and allowed for early detection of various diseases.

Brain cancer is one of the most common types of cancer, affecting millions of people each year. Early detection of brain cancer is crucial in preventing serious complications and potentially saving lives. The current standard method for diagnosing brain cancer involves the expertise of pathologists. However, this approach has its limitations, such as a shortage of skilled pathologists, time-consuming processes, and the potential for misdiagnosis due to fatigue.

To address these challenges, computer-aided brain cancer multiclassification systems have emerged as a promising solution. These systems utilize advanced algorithms and machine learning techniques to assist pathologists in analyzing medical images and improving the accuracy and efficiency of brain cancer diagnosis. By reducing the workload of pathologists and minimizing the risk of misdiagnosis, computer-aided systems can have a significant impact on improving patient outcomes.

I.2Problematic

To effectively treat brain cancer, which is a serious health risk, a prompt and precise diagnosis is essential. Traditional diagnostic methods rely on physicians interpreting imaging scans such as Magnetic Resonance Imaging (MRI) and doing tests. MRI plays a pivotal role in the evaluation of brain tumors as it provides detailed images of the brain's structure and allows for the identification and characterization of abnormalities. These techniques, nevertheless, can be laborious, arbitrary, and prone to mistakes. This thesis suggests a method that makes use of deep learning technology for classifying brain tumors in order to overcome these difficulties. We hope to create a trustworthy and effective system that can reliably distinguish between those with brain cancer and those who do not by utilizing these cutting-edge methodologies.

I.3Motivations

The potential of using Machine Learning in brain tumor detection and classification is immense. It can significantly reduce the time required for diagnosis, enabling prompt intervention and treatment planning. Moreover, it has the potential to enhance the accuracy and consistency of diagnoses, minimizing human errors and subjective interpretations. By leveraging the capabilities of Machine Learning, we aimed to alleviate the burdens and difficulties faced in the medical field when it comes to brain tumor diagnosis.

One effective Machine Learning approach for brain tumor classification is Convolutional Neural Networks (CNNs) excel at extracting features from MRI scans, making them effective for brain tumor classification

Another promising approach is the use of Convolutional Autoencoders(CAEs),CAEs process medical images quickly and detect anomalies like tumors. By combining CNNs and CAEs, we achieve high performance and rapidity in brain tumor detection. This streamlined approach reduces time and effort in the diagnostic workflow, improving efficiency.

I.4Contributions

The thesis segment proposes an approach that incorporates deep learning techniques to classify brain cancer. The objective is to create a dependable and effective system capable of accurately distinguishing between individuals with brain cancer and those who do not have it. By harnessing the power of advanced deep learning techniques, the aim is to develop a robust solution for brain cancer classification.

To achieve our objectives, we employed the following methodology:

1. Convolutional Neural Network (CNN) Foundation: We initially utilized a CNN as the foundation of our approach. Pre-trained models like ResNet50, VGG16, and DenseNet121 were leveraged to extract relevant features from our data.
2. Feature Extraction: The pre-trained CNN models were employed to extract discriminative features from the input data, including both ResNet50 and CAE (Convolutional Autoencoder) models.
3. Support Vector Machine (SVM) Training: Based on the extracted features from ResNet50 and CAE, SVM models were constructed for training. SVMs are widely used classifiers known for their effectiveness in handling high-dimensional data.

4. Fusion Technique: To integrate the results obtained from each SVM, we applied a fusion technique. This process combines the outputs from the ResNet-50 and CAE based SVM models, enhancing the overall classification accuracy.

I.5Thesis structure

The thesis is structured as follows:

- Chapter II: Background

  This chapter offers a comprehensive introduction to the research, covering the fundamental concepts such as digital images, artificial intelligence, deep learning, various types of medical images, and convolutional neural networks,ect.

- Chapter III: Materials and Methods

  In this chapter, the materials and methodologies employed in the study are presented. The focus is on utilizing CNN, VGG-16, ResNet-50, DenseNet-121, and CAE as the key techniques for analysis and classification.

- Chapter IV: Results and Conclusion

  The final section of this dissertation presents the culmination of the research process, encapsulating the findings, analysis, and interpretation of the results obtained from the application of the proposed methodology. Moreover, a comprehensive conclusion is offered, summarizing the entirety of the dissertation.

ChapterII

Work background

II.1Introduction

Artificial intelligence (AI) has emerged as a significant asset in the field of medical imaging, revolutionizing the way clinicians detect and diagnose diseases. Extensive research has demonstrated the potential of AI, particularly through artificial neural networks, to match or even surpass the effectiveness of human radiologists in identifying signs of cancers and various other conditions.

This chapter aims to present the foundational context for our work, encompassing key aspects such as digital image processing, deep learning, different types of deep learning algorithms, and an overview of Convolutional Neural Networks (CNNs).

II.2What is Artificial intelligence?

While a number of definitions of artificial intelligence (AI) have surfaced over the last few decades, [1]offers the following definition:

“AI, or artificial intelligence, is a discipline encompassing the scientific and engineering efforts aimed at creating intelligent machines, particularly intelligent computer programs. It shares a connection with the parallel objective of utilizing computers to comprehend human intelligence, but AI is not restricted to approaches that solely mimic biological observations”.

This paragraph means that Artificial intelligence (AI) is a multidisciplinary field focused on creating intelligent machines and developing computer programs that exhibit intelligent behaviour. One aspect of AI involves understanding and simulating human intelligence, aiming to comprehend how humans perceive, learn, reason, and make decisions. In summary, while AI does study human intelligence to some extent, its scope extends beyond mimicking biological observations. It encompasses scientific and engineering efforts to create intelligent machines and computer programs that exhibit intelligent behavior, leveraging various techniques and disciplines to achieve this goal.

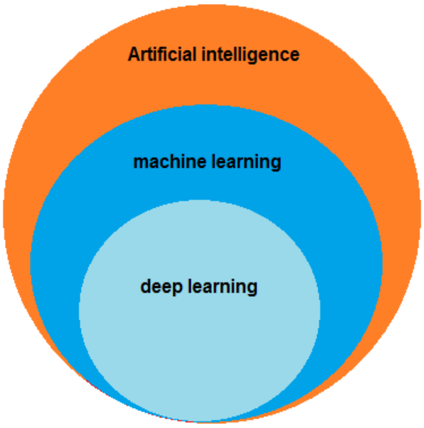


Figure 1 : *the relation between AI,ML, and DL.*

II.3Machine learning

Machine learning has emerged as a crucial subject for development organizations seeking inventive approaches to harness their data resources and enhance their understanding of business operations. Unlike traditional programming methods, machine learning empowers systems to learn from data. Nonetheless, machine learning is a complex undertaking that involves utilizing diverse algorithms to iteratively learn from data, enhance data descriptions, and make predictions. By feeding training data into these algorithms, more accurate models can be generated. These machine learning models represent the outcomes obtained when training a machine learning algorithm with data.[2]

II.3.1Machine learning approaches

In the following, we provide more details on the different ML approaches

II.3.1.1Supervised Learning

In supervised learning, the algorithm learns from labeled training data, where each data point is associated with a corresponding label or target value as depicted in (Fig 2) , examples of supervised learning algorithms include linear regression, decision trees, random forests, support vector machines (SVM), and neural networks.

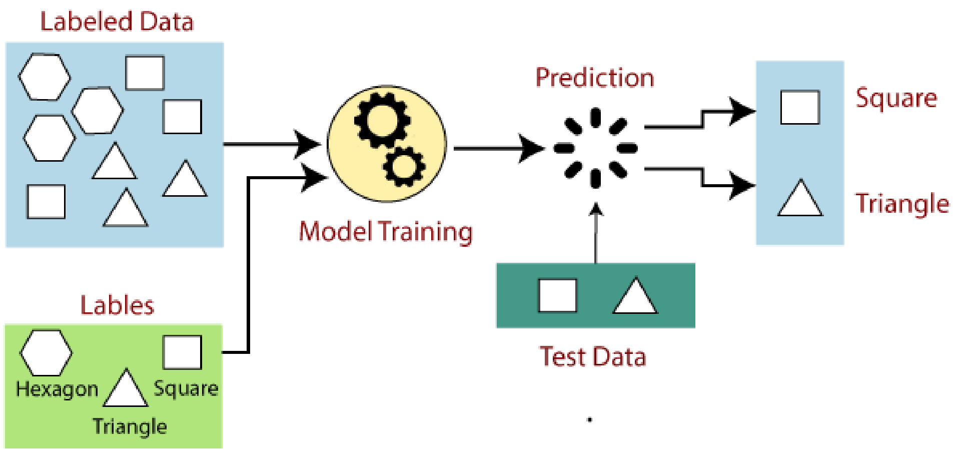


Figure 2 : *supervised machine learning.*

II.3.1.2Unsupervised Learning

Unsupervised learning deals with unlabeled data, where the algorithm learns to find patterns or structure in the data without any specific guidance. Such as k-means and hierarchical clustering, and dimensionality reduction techniques, such as principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE) (fig 3).

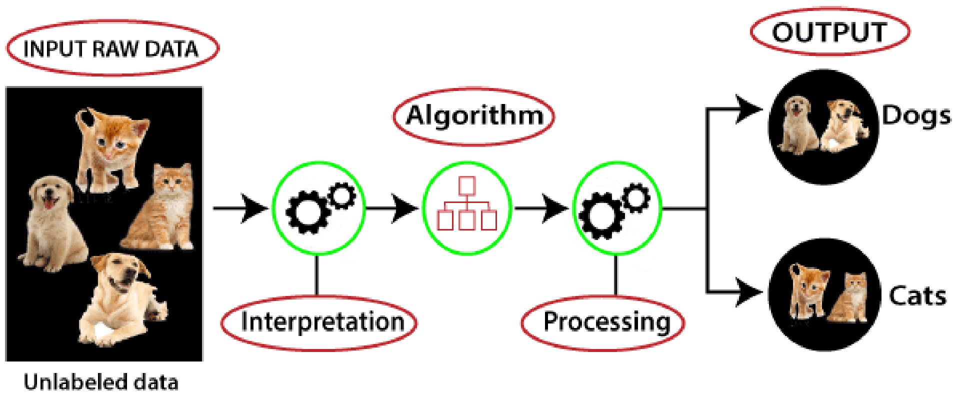


Figure 3 : *unsupervised learning.*

II.3.1.3Semi-supervisedLearning

Semi-supervised learning combines labeled and unlabeled data to improve the learning process. By leveraging the additional information from unlabeled data, it reduces labeling effort, enhances model generalization, and achieves better results in scenarios where labeled data is limited or costly to obtain.[3]

II.4Deep Learning

Deep learning has emerged as a powerful approach for modeling complex data through intricate architectures that incorporate non-linear transformations. Neural networks, including deep neural networks, serve as the fundamental components of deep learning. These techniques have achieved remarkable progress in various domains such as sound and image processing, enabling tasks like facial recognition, speech recognition, computer vision, language processing, and text classification. The potential applications of deep learning are vast and continue to expand.

Different types of neural network architectures, such as multilayer perceptrons, Convolutional Neural Networks (CNNs), and recurrent neural networks, cater to specific data types and tasks. These architectures are characterized by deep layers organized in a cascading manner. Successful implementation of deep learning requires well-designed stochastic optimization algorithms, appropriate initialization techniques, and thoughtful structure selection.

While deep learning yields impressive results, it is important to acknowledge that its theoretical foundations are still relatively limited at present. researchers are continually exploring and advancing the theoretical aspects to further understand and improve deep learning algorithms.[4]

II.5Digital image

To define digital images , we took their definition from a bookUnderstanding Digital Image (Fig 4)[5]:

"A digital image is a two-dimensional function f(x, y) that is a projection of a 3-dimesional scene into a 2-dimensional projection plane, where x, y represents the location of the picture element or pixel and contains the intensity value. When values of x, y and intensity are discrete, then the image is said to be a digital image. Mathematically, a digital image is a matrix representation of a two-dimensional image using a finite number of points cell elements, usually referred to as pixels. Each pixel is represented by numerical values".

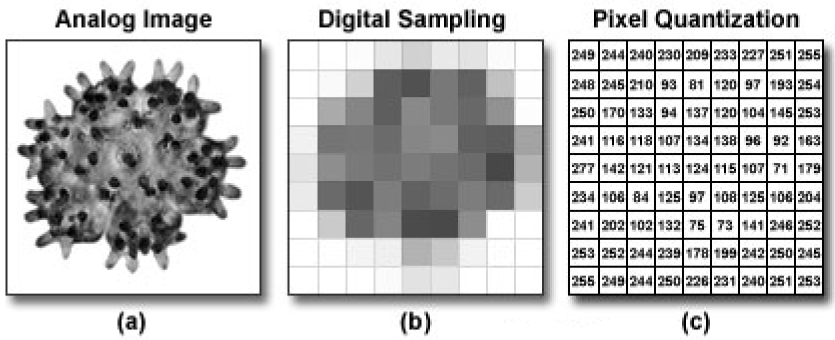


Figure 4 : *digital image.*

II.6Applications of machine learning(ML) and deep learning(DL) in digital imaging

Machine learning (ML) and deep learning (DL) have found numerous applications in the field of digital imaging. Here are a few examples:

- Image Classification: ML and DL algorithms are widely used for image classification tasks. They can automatically analyze and categorize images into different classes or labels. For example, ML models can classify medical images into various disease categories, such as different types of tumors or abnormalities.
- Object Detection: ML and DL algorithms excel in detecting and localizing objects within images. In digital imaging, these algorithms can identify and outline specific structures or objects of interest. This capability is valuable in medical imaging for identifying anatomical landmarks or locating abnormalities.
- Image Segmentation: ML and DL techniques are employed for image segmentation, which involves dividing an image into meaningful regions or segments. This is useful for extracting specific structures or objects from images, such as segmenting tumors or lesions in medical images.

II.7Medical Imaging

Medical imaging encompasses a range of techniques and procedures employed to produce images of the human body, or specific body parts, serving various clinical objectives such as medical procedures, diagnosis, and the study of normal anatomy and function in medical science. In a broader context, it is considered a subset of biological imaging.[6]

II.7.1Type of Medical Imaging

Here are some of the most common types of medical imaging:

II.7.1.1 X-Ray

In X-ray imaging, electromagnetic radiation is utilized to produce images of bones and some organs, as depicted in Figure 5. This imaging technique is commonly employed to detect anomalies such as tumors, infections, and fractures.

Figure 5 : *X-ray. [7]*

II.7.1.2 Computed Tomography(CT)

Computed tomography (CT) scans, alternatively referred to as CT scans, utilize a blend of X-ray technology and computer processing to produce intricate and detailed cross-sectional images of the human body (Fig 6). By combining these two techniques, CT scans provide valuable and comprehensive information regarding the composition and structure of bones, organs, and soft tissues. This imaging modality plays a crucial role in the diagnosis and evaluation of a wide range of medical conditions, including cancer, cardiovascular disease, and internal injuries. The high level of detail offered by CT scans enables healthcare professionals to accurately identify and assess these conditions, thereby facilitating precise treatment and care.

Figure 6 : *Computed Tomography.[8]*

II.7.1.3 Magnetic Resonance Imaging

Magnetic resonance imaging (MRI) uses a powerful magnetic field and radio waves to generate detailed and high-resolution images of organs and tissues inside the human body, as depicted in (Fig7). This non-invasive imaging method offers significant advantages in visualizing intricate details of the brain, spinal cord, joints, and soft tissues. MRI plays a vital role in diagnosing and evaluating a wide range of medical conditions, such as tumors, neurological disorders, and musculoskeletal injuries. Its ability to produce precise and detailed images aids healthcare professionals in accurately identifying and assessing these conditions, enabling effective treatment planning and optimal patient care.[9]

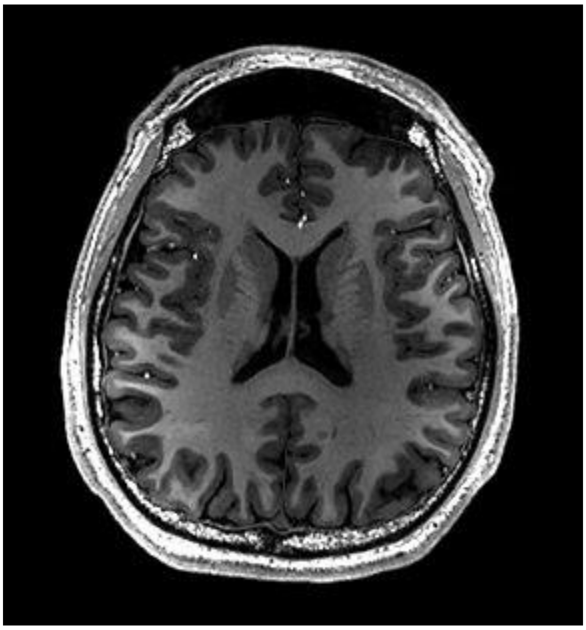


Figure 7 : *Magnetic Resonance Imaging. [10]*

II.7.1.4 Mammography

Mammograms are specific X-ray screenings performed on the breast to detect and diagnose breast diseases, with a primary focus on breast cancer. They are specialized examinations aimed at identifying potential abnormalities and providing early detection of breast conditions. Mammograms play a crucial role in the detection, diagnosis, and monitoring of breast cancer, enabling healthcare professionals to initiate timely and appropriate treatment plans. This imaging technique is an essential tool in promoting breast health and improving outcomes for individuals at risk of or affected by breast diseases.

In the following section, we will explore the concepts and advancements in neural networks, particularly CNNs.

II.8 Artificial Neural Networks

Artificial Neural Networks (ANNs) are algorithms that draw inspiration from the functioning of the brain and are employed for modeling complex patterns and making forecasts. ANNs, as a deep learning method, emerged from the concept of Biological Neural Networks in the human brain, aiming to recreate their operations. While ANNs exhibit significant similarities to biological neural networks, they are not exact replicas. ANNs are specifically designed to process numeric and structured data, accepting inputs of that nature exclusively.[11]

II.8 .1The architecture of neural networks

The leftmost layer in this network is called the input layer, and the neurons within the layer are called input neurons. The rightmost or output layer contains the output neurons, or, as in this case, a single output neuron. The middle layer is called a hidden layer, since the neurons in this layer are neither inputs nor outputs. The network above has just a single hidden layer, but some networks have multiple hidden layers. For example, the following figure (fig 8) represents a four-layer network with two hidden layers.

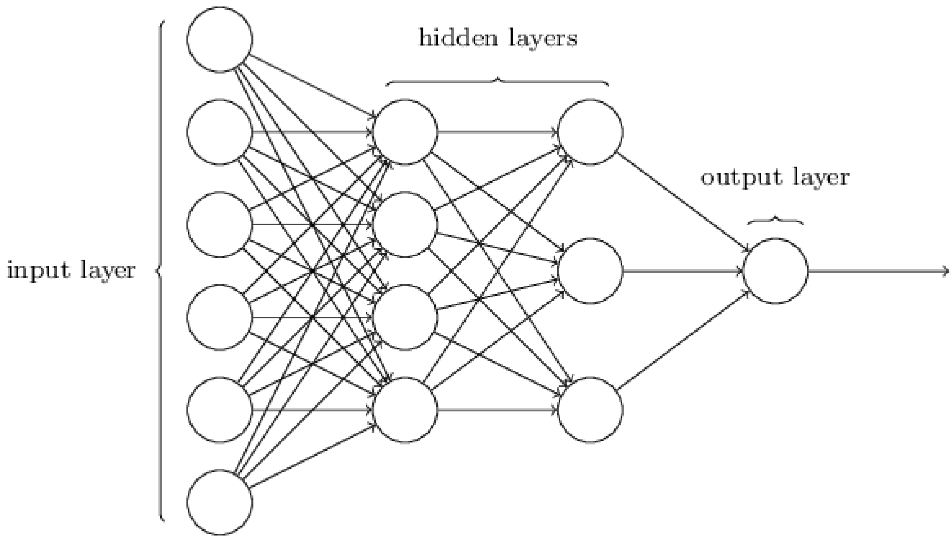


Figure 8 : *general structure of an ANN.*

II.9 Convolutional neural network

Convolutional Neural Networks (CNNs) are deep learning algorithms that utilize multi-layer neural networks to learn important features from images. They have the ability to perform various tasks, including object classification (Fig 9), detection, and segmentation, by leveraging these learned features**.**[12]

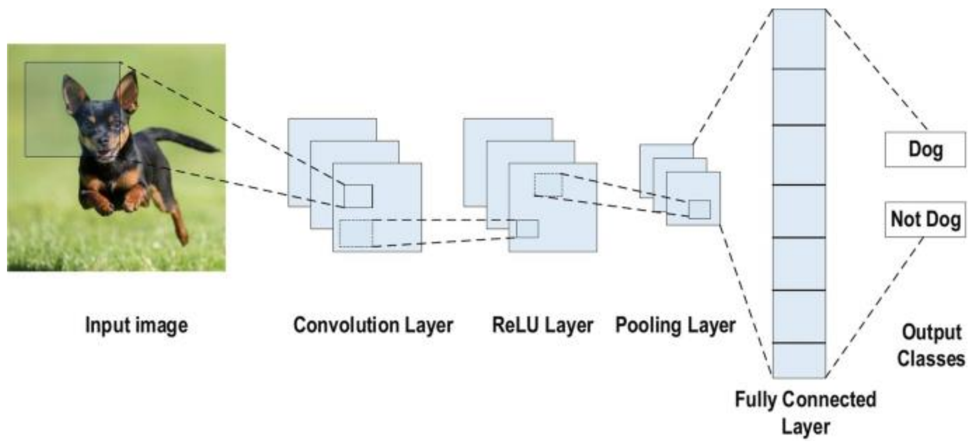


Figure 9: *Convolutional Neural Network architecture for image classification.[13]*

II.9.1 Basic architecture of CNN

The basic architecture of CNN is composed of the following layers :

a- Convolution layer

b- RELU layer

c- Pooling layer

d- Fully connected layer

e- Output layer (soft-max)

II.9.1.1 Convolution layer

- A convolutional layer is composed by a set of filters, also called kernels that slides over the input data(Fig 10).
- Each kernel has a width, a height and width x height weights utilized to extract features from the input data.
- In the training step, the weights in the kernel starts with random values, and will be learning based on the training set.
- Each filter in the convolutional layer represents a feature.[14]

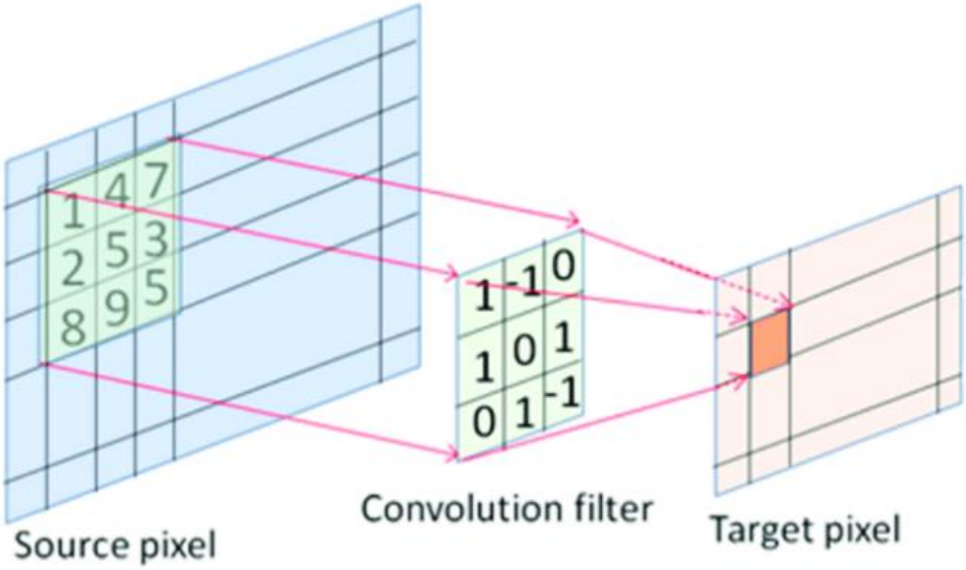


Figure 10 : *Convolution layer.*[15]

II.9.1.2 RELU layer

In this layer every negative value from the filtered image is removed and replaced with zero (Fig 11).

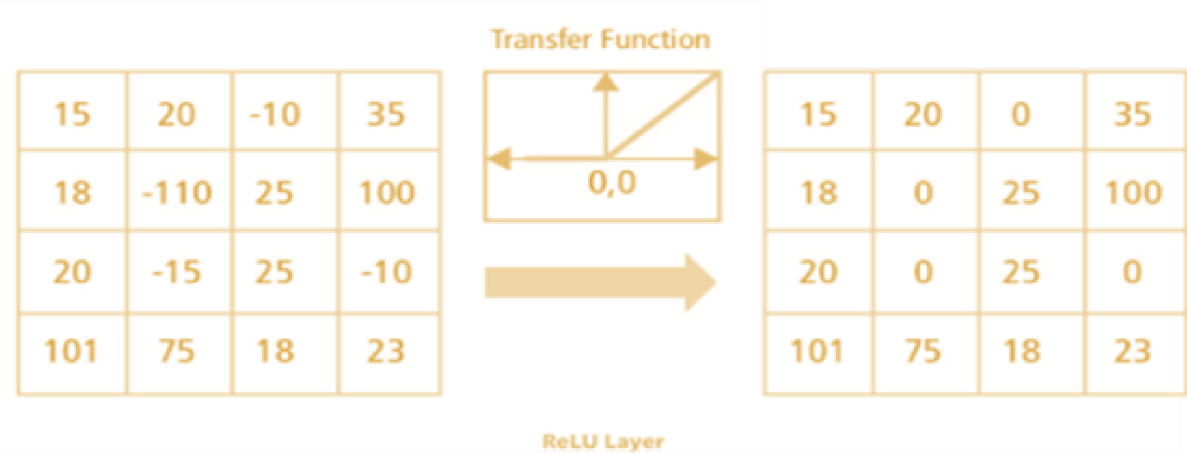


Figure 11 : *Relu operation.[16]*

II.9.1.3 Pooling layer

The pooling layer, also known as the down-sampling layer, is utilized to decrease the dimensionality of feature maps while retaining the most important information. Within the pooling layer, a filter moves across the input data and performs a pooling operation, such as maximum, minimum, or average pooling (Fig 12). Among these options, max pooling is commonly favored in the literature.[14]

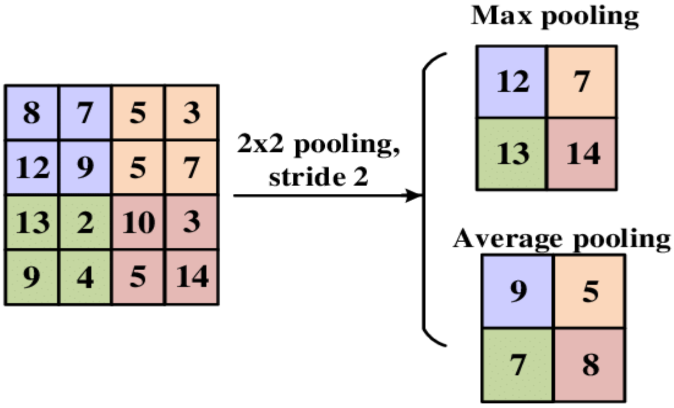


Figure 12 : *Pooling Layer. [17]*

II.9.1.4 Fully-Connected Layer

The fully-connected layer, as its name suggests, is responsible for performing classification based on the features extracted by the preceding layers and their respective filters. This layer connects every neuron from the previous layer to every neuron in the current layer, creating a dense network of connections. By leveraging these connections, the fully-connected layer combines and processes the extracted features to make predictions or classifications on the input data (Fig 13). In this way, it considers the collective information learned from the earlier layers to perform the final classification task.[18]

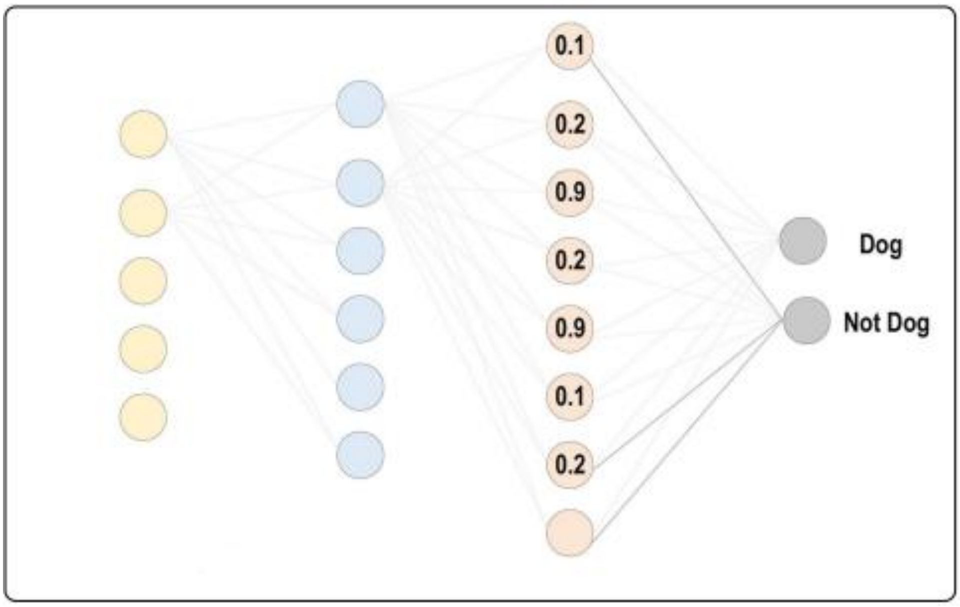


Figure 13 : *fully connected layer.[13]*

II.9.1.5 Outputlayer

In neural networks, the output layer is the last layer of neurons that produces the final output of the network for a given input. The number of neurons in the output layer of a neural network is determined by the specific task that the network is designed to perform.

II.10 Convolution autoencoder

Autoencoders are unsupervised generative models in which an artificial neural network (ANN) is trained to reconstruct its own input. They consist of two main components: an encoder and a decoder. The encoder compresses the input data into a lower-dimensional representation that captures the essential information. The decoder, on the other hand, reconstructs the input from the encoded features. After unsupervised pretraining, the encoderbecomes an effective automatic feature extractor. By adding an appropriate output layer, the encoder can be fine-tuned in a supervised manner to achieve the desired estimation. [19]

The autoencoder consist of three parts: an encoder, a decoder and bottleneck (Fig 14):

II.10.1Encoder

The first section of an autoencoder, the encoder is the convnet that acts specifically as a feature extractor. Its primary function is to help extract the most salient features from images and return them as a vector.

II.10.2 Bottleneck

Located right after the encoder, the bottleneck, also called a code layer, serves as an extra layer which helps to compress the extracted features into a smaller vector representation. This is done in a bid to make it more difficult for the decoder to make sense of the features and force it to learn more complex mappings.

II.10.3 Decoder

The last section of an autoencoder, the decoder is that convnet which attempts to make sense of the features coming from the encoder, which have been subsequently compressed in the bottleneck, so as to reconstruct the original image as it was.[20]

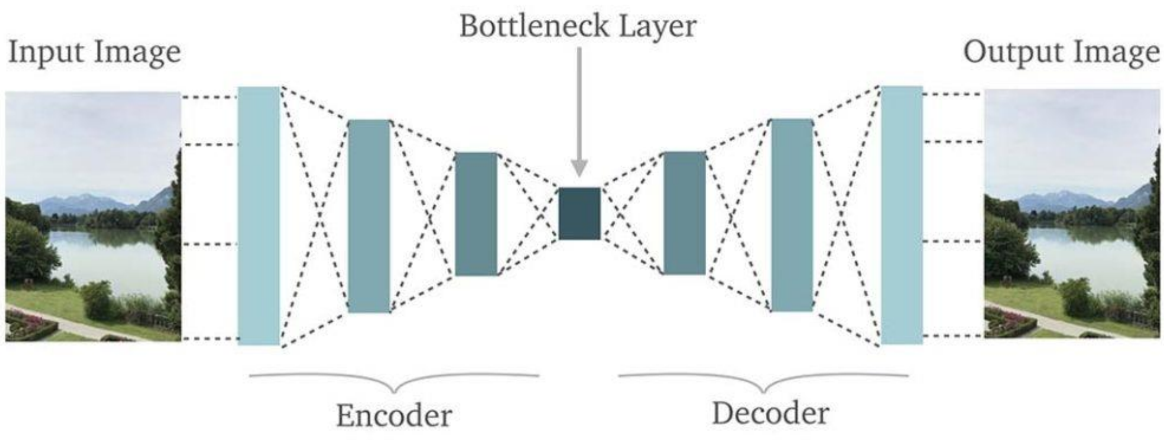


Figure 14 : *Convolutional autoencoder. [21]*

II.11 Conclusion

In this chapter, we have given an overview of some important concepts, starting with the digital image, processing and deep learning, passing through a convolutional neural network and their architecture, finalizing by auto-encoder. Our goal was to present some details about the background of the work, making it easier to understand the work and its context with clarity.

ChapterIII

Material and Methods

III.1 Introduction

The aim of this chapter is to present the suggested techniques and lay out the methodology used in our research. Our work spans two different architectural styles. To extract features from the dataset for the first architecture, we used pre-trained models like ResNet-50, VGG-16, and DenseNet-121. These models were effective feature extractors because they had already been trained on significant picture classification tasks. We then used a fusion model to fuse these extracted characteristics, allowing us to combine the advantages and variety of representations of the pre-trained models. With the second design, we concentrated on fusing a residual network (resent) with a unique model known as CAE (Classification and Evaluation design). By combining these two models' distinct strengths, the goal was to improve classification performance.

III.2 Proposed solution (scheme 1)

We used VGG-16, ResNet-50, and DenseNet-121 models in this architecture to extract features from the input data. Then, in order to classify data, we trained these models using SVM (Support Vector Machine). The results of the SVM models were then integrated using a fusion technique. As depicted in (Figure 15).

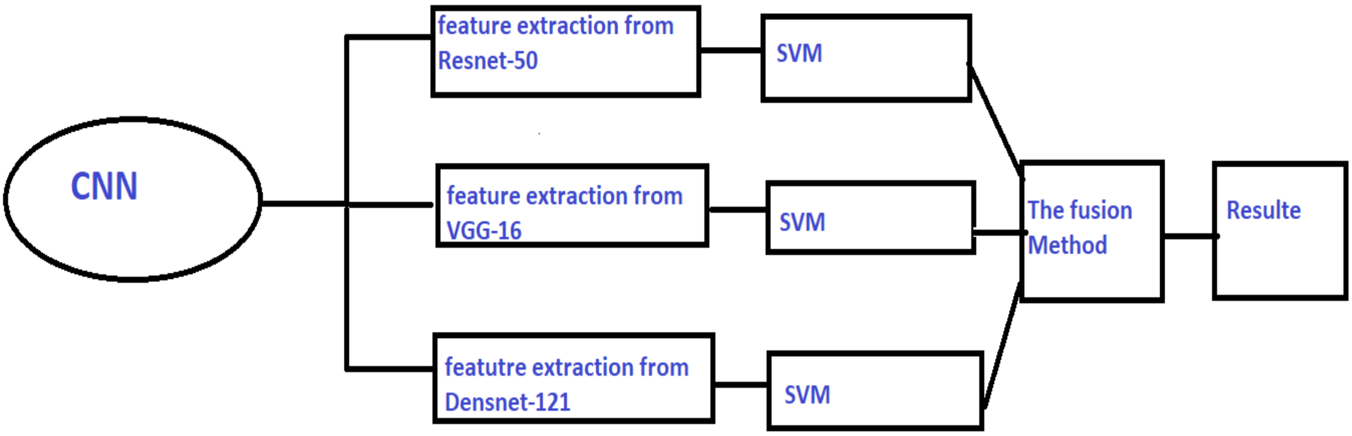


Figure 15:The firstArchitecture of our methods.

We will explain each element in the previous diagram in the following

III.2.1 Convolutional neural networks

In this section, we introduced a convolutional neural network (CNN) model specifically designed for image classification tasks. The model comprises multiple layers, each serving a specific purpose in the classification process. Additionally, we specified a certain number of epochs to train the model, allowing it to learn and optimize its performance over a set number of iterations.

III.2.2 Activation function used

III.2.2.1 Rectified Linear unit (ReLU)

ReLU is an activation function commonly used in Convolution Neural Network (CNNs). It will be applied to point where the negative ones are converted to zero (Fig16).

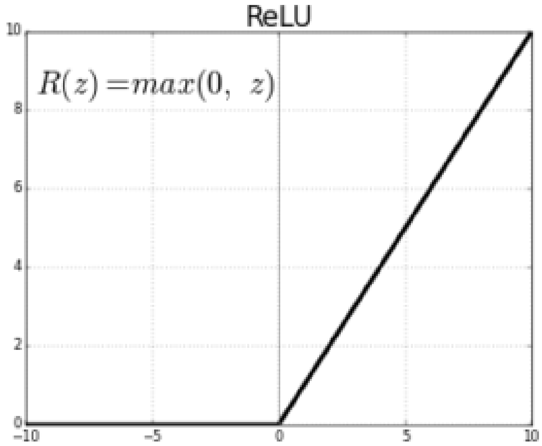


***Figure 16 :** ReLUactivation function.[22]*

III.2.2.2 SoftMax function

- The softmax function is a mathematical function that is commonly used in machine learning, specifically in multi-class classification problems. It takes as input a vector of real numbers and transforms them into a probability distribution, where each element of the vector represents the probability of belonging to a particular class.
- The softmax function is often used as the final activation function in the output layer of a neural network when performing multi-class classification.[4]

III.2.3 Loss function used

III.2.3.1 Categorical cross-entropy

The Categorical Cross-Entropy is a widely used loss function in machine learning for multi-class classification problems. It compares the predicted probabilities of different classes with the true class labels. By measuring the dissimilarity between the predicted probability distribution and the true distribution, Categorical Cross-Entropy provides a quantification of the error between the predictions and the actual classes. This loss function is commonly paired with softmax activation at the output layer of neural networks to convert raw outputs into a valid probability distribution. Minimizing the Categorical Cross-Entropy loss during training helps the model improve its ability to accurately classify inputs into the correct classes.[23]

III.2.5 Resnet50

ResNet50 is a deep convolutional neural network architecture that was introduced by Microsoft Research in 2015. The name ResNet-50 stands for “Residual Network”, as the network is built using residual blocks, which allow for the training of very deep networks without the vanishing gradient problem.[24]

III.2.5.1 Resnet50 architecture

ResNet-50 is a deep neural network architecture consisting of 50 layers. It is composed of multiple residual blocks, which include convolution layers, batch normalization layers, and shortcut connections. These shortcut connections allow gradients to flow more effectively during training, addressing the vanishing gradient problem. The architecture begins with a convolution layer, followed by a batch normalization layer, a ReLU activation layer, and a max pooling layer. It then repeats the residual blocks, each containing convolution and batch normalization layers, as well as shortcut connections. The weights and shortcut connections vary between these blocks. The repetition of these blocks continues until the final output layer is reached. The last layer of ResNet-50 is a fully connected layer with 1000 units, corresponding to the number of classes in the ImageNet dataset (fig 17).[25]

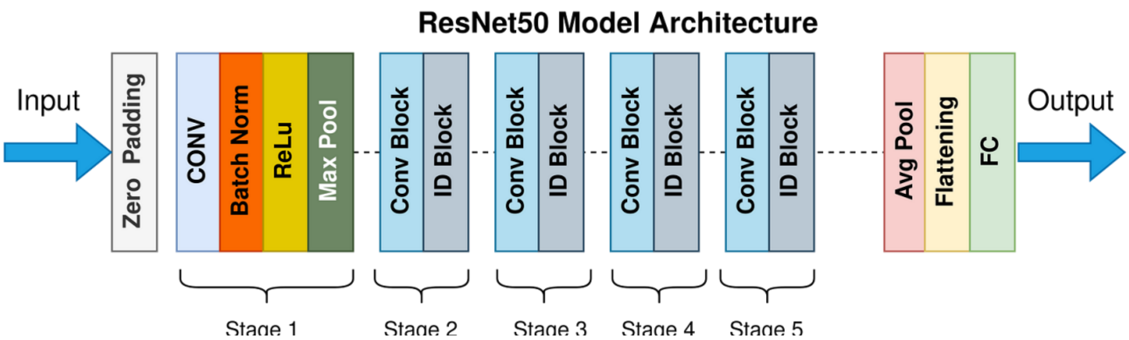


Figure 17 : *Architecture Resnet-50.*

III.2.6 VGG16

VGG16 is a convolutional neural network architecture that was introduced by the Visual Geometry Group (VGG) at the University of Oxford in 2014. It consists of 16 layers of convolutional and fully connected layers.[26]

The VGG16 architecture consists of 13 convolutional layers, followed by 3 fully connected layers. The convolutional layers are designed to learn hierarchical representations of the input images, starting from simple low-level features such as edges and textures, and gradually increasing in complexity to learn more high-level features such as object parts and shapes.

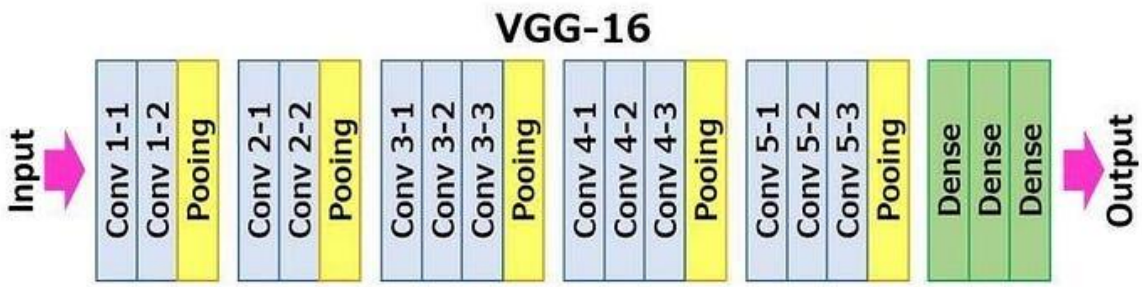


Figure 18:*Architecture VGG-16. [25]*

III.2.7 Densenet-121

DenseNet-121 is a type of convolutional neural network architecture that utilizes dense connections between layers. In a dense block, each layer is directly connected to every other layer within the block, forming a dense connection pattern. This connectivity enables each layer to receive feature maps from all preceding layers in the block.[27]

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	[1 × 1 conv, 3 × 3 conv] × 6	[1 × 1 conv, 3 × 3 conv] × 6	[1 × 1 conv, 3 × 3 conv] × 6	[1 × 1 conv, 3 × 3 conv] × 6
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	[1 × 1 conv, 3 × 3 conv] × 12	[1 × 1 conv, 3 × 3 conv] × 12	[1 × 1 conv, 3 × 3 conv] × 12	[1 × 1 conv, 3 × 3 conv] × 12
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	[1 × 1 conv, 3 × 3 conv] × 24	[1 × 1 conv, 3 × 3 conv] × 32	[1 × 1 conv, 3 × 3 conv] × 48	[1 × 1 conv, 3 × 3 conv] × 64
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	[1 × 1 conv, 3 × 3 conv] × 16	[1 × 1 conv, 3 × 3 conv] × 32	[1 × 1 conv, 3 × 3 conv] × 32	[1 × 1 conv, 3 × 3 conv] × 48
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

The layers in the table are as follows:

1. Basic convolution layer with 64 filters of size 7X7 and a stride of 2
2. Basic pooling layer with 3x3 max pooling and a stride of 2
3. Dense Block 1 with 2 convolutions repeated 6 times
4. Transition layer 1 (1 Conv + 1 AvgPool)
5. Dense Block 2 with 2 convolutions repeated 12 times

6. Transition layer 2 (1 Conv + 1 AvgPool)
7. Dense Block 3 with 2 convolutions repeated 24 times
8. Transition layer 3 (1 Conv + 1 AvgPool)
9. Dense Block 4 with 2 convolutions repeated 16 times
10. Global Average Pooling layer- accepts all the feature maps of the network to perform classification
11. Output layer

Therefore, DenseNet-121 has the following layers:

- 1 7x7 Convolution
- 58 3x3 Convolution
- 61 1x1 Convolution
- 4 AvgPool
- 1 FullyConnected Layer.[27]

III.2.8Support Vector Machine(SVM)

Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression .They belong to a family of generalized linear classifiers. In another terms, Support Vector Machine (SVM) is a classification and regression prediction tool that uses machine learning theory to maximize predictive accuracy while automatically avoiding over-fit to the data. Support Vector machines can be defined as systems which use hypothesis space of a linear functions in a high dimensional feature space, trained with a learning algorithm from optimization theory that implements a learning bias derived from statistical learning theory.[28]

III.2.8.1 Linaire support vector machine :

When the data is perfectly linearly separable, it means that the data points can be divided into two distinct classes using a single straight line (in the case of 2D data) (Fig 19). In such cases, Linear Support Vector Machine (SVM) can be effectively used for classification.[29]

This picture will show how it works:

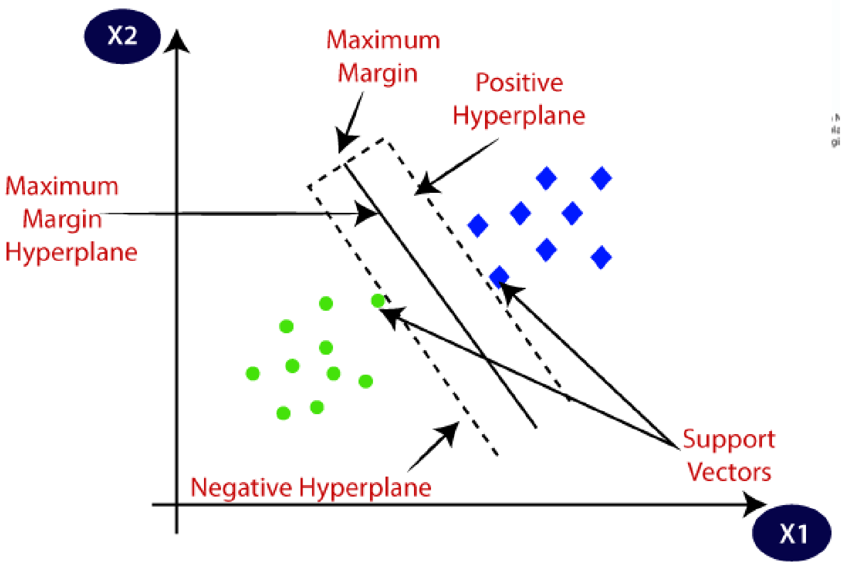


Figure 19 : *Architecture of linear support vector machine.*[29]

III.2.9The fusion technique

This section delves into the application of fusion methods in our models.

III.2.9.1 feature-level fusion

For image processing, this technique combines characteristics from three different models, ResNet-50, VGG-16, and DenseNet-121.a collection of models is trained using these characteristics after the necessary data has been collected, and the training is carried out using an SVM model with a linear kernel. By combining the strengths of several models and using a linear SVM for training, this method seeks to improve the efficiency and accuracy of image analysis.

III.2.9.2 decision-level fusion

This method utilizes data fusion, which involves combining information from multiple sources to reach a unified conclusion or prediction. In this case, we merge the output labels from the ResNet-50, VGG-16, and DenseNet-121 models. The objective is to leverage the collective wisdom of these models to make a more accurate prediction. The labels generated by each model are used to vote on the potential labels associated with the image, enabling us to arrive at a final decision based on a consensus among the models. This approach harnesses the power of ensemble learning and data fusion to enhance the robustness and reliability of the predictions made by the individual models.

III.2.9.3 score-level fusion

In this method, we consider the probabilities generated by ResNet-50, DenseNet-121, and VGG-16 models. We symbolize the output probability of each model as $\hat{S}$.
Now, let's examine the formulas for each score level fusion function, which we symbolize as (f_max, f_min,f_product, f_sum, and f_mean):
$\hat{S}_1$:The output of Resnet-50
$\hat{S}_2$:The output of VGG-16
$\hat{S}_3$:The output of Densnet-121
Equation 1:f max=$\sum_{i=1}^{k}(\max(\widehat{S}_1, \hat{S}_2, \hat{S}_3))$
Equation 2:f min=$\sum_{i=1}^{k}(\min(\widehat{S}_1, \hat{S}_2, \hat{S}_3))$
Equation 3:f product= $\prod_{i=1}^{k}(\widehat{S}_1, \hat{S}_2, \hat{S}_3)$
Equation 4:f sum=$\sum_{i=1}^{k}(\widehat{S}_1, \hat{S}_2, \hat{S}_3)$
Equation 5:f mean= $\sum_{i=1}^{k}(\widehat{(S}_1, \hat{S}_2, \hat{S}_3) \div 3)$
In our work, the Mean function is used in our model.

III.3 Proposed solution (scheme 2)

This architecture adopts a multi-step approach for its processing. Firstly, the input data is fed into a pre-trained ResNet-50 model to extract relevant features. These features are then used in conjunction with a custom model known as CAE (Classification and Evaluation Architecture). The CAE model is designed to incorporate the extracted features and further process them.
Following the CAE stage, SVM models are employed for the purpose of classification. SVMs are known for their effectiveness in separating data points into distinct classes based on their features. In this architecture, the SVM models utilize the processed features from the CAE model to classify the input data accurately.
Lastly, a fusion technique is applied to integrate the outputs of the SVM models. This fusion process combines the individual classification results from the SVMs to arrive at a final

prediction or decision. The fusion technique ensures that the collective knowledge and predictions of the SVM models are effectively combined to improve the overall accuracy and robustness of the system. As shown in the figure 20.

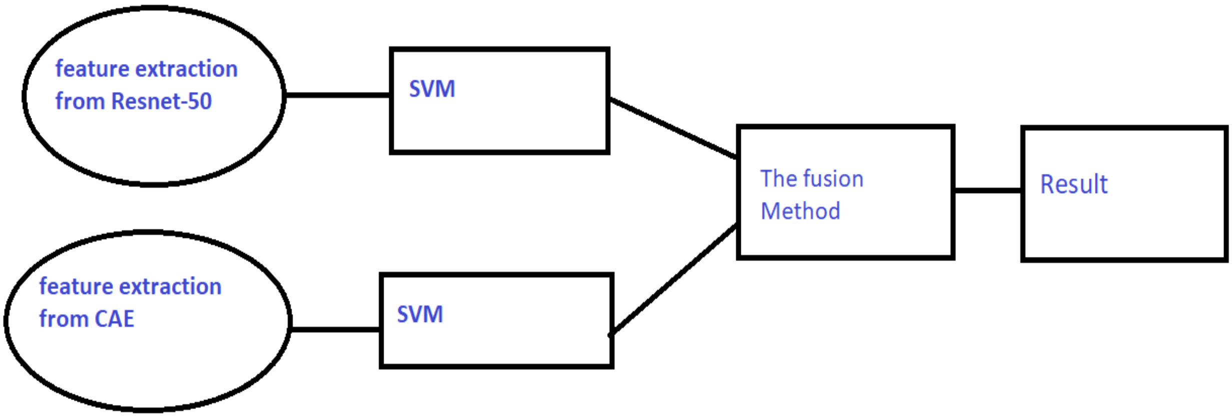


Figure 20 : *Architecture2 of our methods.*

Now we will explain each element in detail

III.3.1 Convolution autoencoder implementation

The primary goal of this study is feature extraction using the Convolutional Autoencoder (CAE) model. The CAE model plays a vital role in reconstructing input images by utilizing the extracted features.

The encoder layer and the decoder layer are the two main parts of the CAE model, often known as the Classification and Evaluation Architecture. Together, these elements encrypt the incoming data into a latent representation that captures the key properties. The decoder layer then reconstructs the original input images by decoding this representation. The encoder and decoder layers of the CAE model can be used to efficiently extract useful features from the input data for additional processing and analysis. In summary, our primary goal is to leverage the CAE model for feature extraction. The CAE model, comprising the encoder and decoder layers, enables us to encode the input data into a meaningful latent representation and then reconstruct the original images using the decoder layer. This approach allows us to extract relevant features from the input data for subsequent analysis and processing.

III.3.1.1 encoder layer

The architecture consists of three convolutional layers, with filter sizes of 128, 64, and 16, respectively. The kernel size for each convolutional layer is 3x3. To ensure that the output size matches the input size, padding is set to 'same'. The activation function used in each layer is the Rectified Linear Unit (ReLU).

Following each convolutional layer, a MaxPooling2D layer with a shape of 2x2 is applied. This layer helps reduce the spatial dimensions of the feature maps, effectively down sampling the information learned by the convolutional layers. The MaxPooling2D layer extracts the maximum value within each 2x2 region, further capturing the most important features in the data.

In summary, the architecture utilizes three convolutional layers with progressively decreasing filter sizes, ReLU activation, and 3x3 kernel sizes. Each convolutional layer is then followed by a MaxPooling2D layer of shape 2x2 to down sample the feature maps.

III.3.1.2 Decoder layer

The architecture consists of three convolutional layers with filter sizes of 16, 64, and 128, respectively. The kernel size for each convolutional layer is 3x3. To maintain the input-output size consistency, padding is set to 'same'. The Rectified Linear Unit (ReLU) activation function is applied to each layer.

Following each convolutional layer, an UpSampling2D layer with a shape of 2x2 is used. This layer helps increase the spatial dimensions of the feature maps, effectively upsampling the information learned by the convolutional layers. The UpSampling2D layer replicates each element within a 2x2 region, thus expanding the feature maps.

In summary, the architecture comprises three convolutional layers with increasing filter sizes, ReLU activation, and 3x3 kernel sizes. Each convolutional layer is then sequentially followed by an UpSampling2D layer of shape 2x2, allowing for upsampling and expansion of the feature maps.

Finally, wegenerate a summary of the model as shown in Table 1:

Layer (type)	Output Shape	Parameters
INPUT (InputLayer)	[(None, 128, 128, 3)]	0
conv2d_12 (Conv2D)	(None, 128, 128, 128)	3584
max_pooling2d_8(MaxPooling2D)	(None, 64, 64, 128,)	0
conv2d_13 (Conv2D)	(None, 64, 64, 64)	73792
max_pooling2d_9(MaxPooling2D)	(None, 32, 32, 64)	0
conv2d_transpose_6 (Conv2DTranspose)	(None, 16, 16, 16)	2320
up_sampling2d_6 (UpSampling2D)	(None, 32, 32, 16)	0
conv2d_transpose_7Transpose)	(None, 32, 32, 64)	9280
up_sampling2d_7 (UpSampling2D)	(None, 64, 64, 64)	0
conv2d_transpose_8 (Conv2DTranspose))	(None, 64, 64, 128)	73856
up_sampling2d_8 (UpSampling2D)	(None, 128, 128, 128)	0
OUTPUT (Conv2D)	(None, 128, 128, 3)	3459
Total params	175,523	
Trainable params	175,523	
Non-trainable params	0	

Table 1 : *Model convolution autoencoder.*

III.3.2Resnet-50

In this section, we will delve into the practical steps where the implementation of the discussed architecture in section III.2.5is illustrated.

III.3.2.1 Resnet50 implementation

In our implementation of Resnet-50, the following settings are used

- **Weights="imagenet”:** refers to a set of pre-trained parameters that have been learned from the ImageNet dataset, which can be used as a foundation for various computer vision tasks.
- **include_top=False:** this parametermeans that the final layer(s) of a pre-trained model are not included during prediction, allowing you to use the lower layers of the model for feature extraction or as a building block for further customization.
- **input shape=(128, 128, 3).**

III.3.3The fusion technique

In our model, we incorporated the fusion technique to combine the extracted features from ResNet-50 and the trained CAE model, as explained earlier in sections III.2.10. This fusion process involved three different approaches: feature-level fusion, decision-level fusion, and score-level fusion.

Feature-level fusion involved combining the extracted features from ResNet-50 and the CAE model at the feature level. This allowed us to merge the representations learned by both models and create a more comprehensive and informative feature representation.

Decision-level fusion focused on combining the classification decisions made by ResNet-50 and the CAE model. By aggregating the individual decisions, we aimed to improve the overall accuracy and robustness of the classification process.

Score-level fusion involved combining the confidence scores or probabilities generated by ResNet-50 and the CAE model for each class. This fusion technique aimed to enhance the reliability of the classification results by considering the confidence levels from multiple models.

By employing these fusion techniques, we sought to leverage the strengths of ResNet-50 and the CAE model, and create a more powerful and accurate classification system.

III.4 Conclusion

In this chapter, we have (1examined the two different architectures we used and 2) outlined the methodologies adopted. Three well-known CNN models were used in the first architecture: VGG-16, ResNet-50, and DenseNet-121. These models were applied to the input data in order to extract features. The distinct architecture of each model and its pre-trained weights enabled it to capture various facets and representations of the input images. We concentrated on fusing the ResNet-50 model and the CAE model in the second architecture.

We specifically integrated the features produced by the CAE model and those recovered by ResNet-50. SVM models were then used to make predictions in the classification challenge using this combined feature representation. SVMs are a good choice for our assignment because they have a reputation for accurately classifying data items based on their attributes. To improve the performance, a fusion technique is used .in the next chapter, the obtained result for each of the models used in our experiments are provided.

Chapter IV

Experimental Results

IV.1 Introduction

In this chapter, we extensively explore the results obtained from our models as discussed in Chapter 3. Our main objective is to evaluate the effectiveness and performance of the proposed method through a series of experiments. We carefully assess the experimental outcomes from various perspectives and provide a detailed presentation of our findings.

During these experiments, we employ several evaluation metrics to thoroughly analyze the performance of our models. These metrics include accuracy, precision, recall, and F1-score, which enable us to quantitatively measure the models' capabilities in terms of classification accuracy, the ability to correctly identify positive instances, and the ability to minimize false positives and false negatives.

To summarize, this chapter serves as an in-depth analysis of the results obtained from our models. By conducting various experiments and employing appropriate evaluation metrics, we thoroughly assess the proposed method's effectiveness and performance. The findings presented in this chapter provide valuable insights into the models' capabilities and pave the way for drawing informed conclusions about their overall performance.

IV.2 Dataset

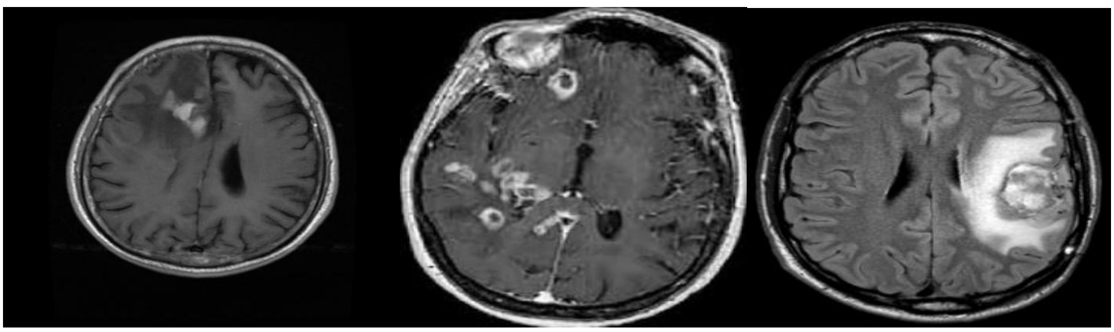
- For our research on ("brain cancer"), we needed highly accurate data. Therefore, weselected,a dataset from the Kaggle website ("https://www.kaggle.com/datasets/preetviradiya/brian-tumor-dataset"), which includes radiographic images of the MRI type captured from various angles to facilitate tumor identification.
- These images are categorized into two classes: Brain tumor, Healthy.
- The dataset comprises a total of 4600 images that have been appropriately divided into their :

The Classes	Number of image
Brain Tumor	**2513**
Healthy	**2087**
Total	**4600**

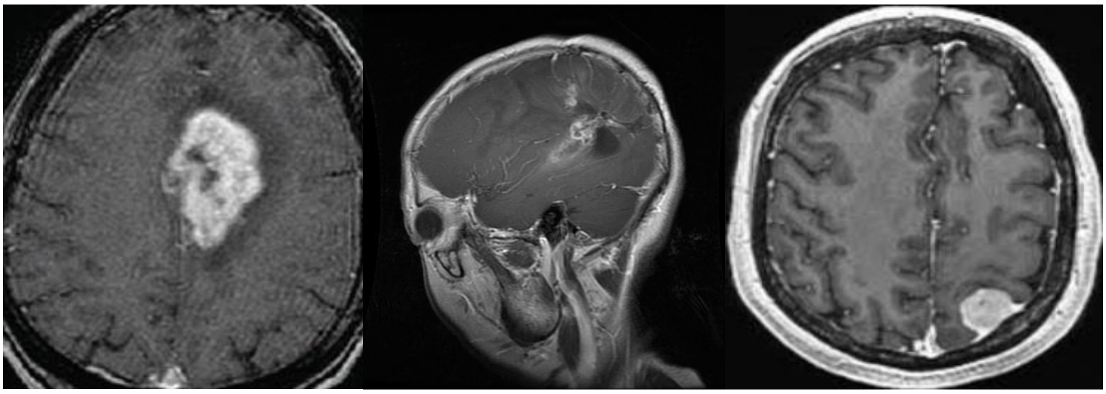
Table 2: *description of the brain tumor dataset.*

IV.3 Data images:

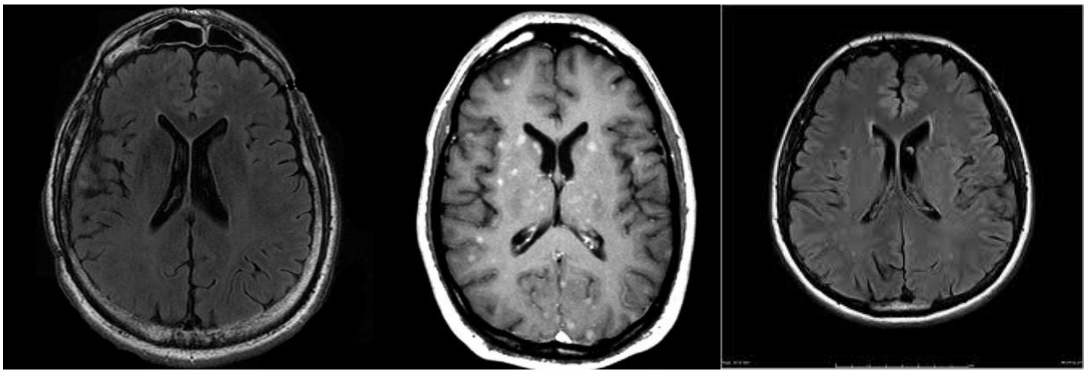
Brain tumor Brain tumor Brain tumor



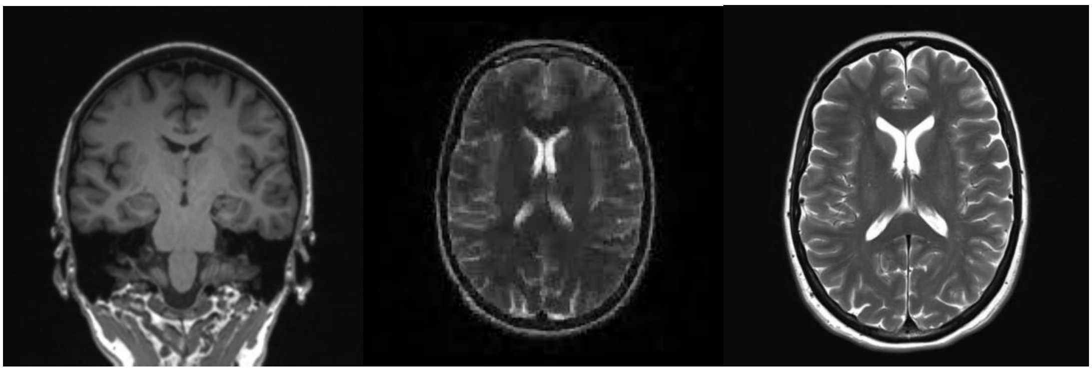
Brain tumor Brain tumor Brain tumor



HealthyHealthyHealthy



HealthyHealthyHealthy



IV.4 Experimental settings

IV.4.1 Libraries

In our implementation, the Python language is used relying on a set of libraries that includes:

- TensorFlow,
- Keras,
- Pandas,
- Sklearn,
- Numpy,
- PIL, and
- MatPlotib.

We decided to use Python as our main programming language to meet our needs for programming. Python is a good choice for jobs involving machine learning and deep learning because to its vast ecosystem of tools and frameworks.

We made use of the Kaggle platform to help our research and development efforts. We had access to a useful tool for building and executing deep learning models thanks to Kaggle. The platform is a perfect fit for our project because it combines strong functionality with free resources.

We were able to take advantage of Kaggle's enormous collection of datasets, pre-trained models, and computational resources by utilizing the tools that are available there. This aided our investigations and made it possible for us to train and test our models quickly.

IV.5 Hardware

We used two different laptops with the specifications presented in Table 3:

PC1	PC2
1.Dell Latitude3350: • Processor: Intel® Core(TM) i5-5200U CPU @2.20Hz @ 2.19Hz. • RAM: 4G RAM. • Operating System: Windows 10. • Graphics Card: Intel (R) HD Graphics 5500.	2.Lenovo-PC : • Processor: Intel® Core(TM) i5-3320M CPU @2.60GHz @ 2.60GHz. • RAM: 4G RAM. • Operating System: Windows 7. • Graphics Card: Intel (R) HD Graphics 4000.

Table 3: *Hardwar used.*

IV.6 The proposed method Results

IV.6.1 Experimental protocol

We divided the dataset into 80% for training and 20% for testing, and we utilized them with our models mentioned in Chapter 3. Now, we will examine the results of each method and the adopted architecture.we identified the dataset as RGB images.

IV.6.2 Performance matrics:

To evaluate the performance of a model, several metrics are considered:

- **Accuracy**

Accuracy is a measure of how close a result or prediction is to the true or expected value. It is commonly used to evaluate the performance of a model or system by comparing the number of correct predictions or outcomes to the total number of predictions made. In other words, accuracy quantifies the degree of correctness or exactness in the output of a process or algorithm. It is often expressed as a percentage, where a higher accuracy value indicates a higher level of precision and reliability in the results

$$\textbf{Equation4:Accuracy} = \frac{\textbf{Number of test images correct classified}}{\textbf{Total Number of test images}}$$

- **Recall**

Recall is a metric that measures the proportion of relevant instances correctly identified by a model. It focuses on minimizing false negatives, ensuring that fewer positive instances are missed. A higher recall value indicates better performance in capturing the positive instances.

$$\textbf{Equation1:Recall} = \frac{\textbf{Sum of all True Positives}}{\textbf{Sum of all True Posotives+Sum of all False Negatives}}$$

- **Precision**

Precision is a performance metric that measures the accuracy of positive predictions made by a model. It quantifies the proportion of correctly predicted positive instances out of all instances predicted as positive.

$$\textbf{Equation2:Precision} = \frac{\textbf{Sum of all True Positives}}{\textbf{Sum of all True Positives+Sum of all False Positives}}$$

- **F1-Score**

The F1 score is a performance metric used in machine learning and statistics that combines the precision and recall metrics into a single value. It provides a balanced measure of a model's accuracy by taking into account both the model's ability to correctly identify positive instances (precision) and its ability to capture all relevant positive instances (recall).

$$\textbf{Equation3:F1} - \textbf{Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

IV.6.3 The Results

In this section the set of the results we obtained for each model we employed. To facilitate a comprehensive understanding of the outcomes, we will present them in the form of tables, matrices, and curves.

These tables will include evaluation measures such as accuracy, precision, recall, and F1-score. The values for each metric will be presented for different experiments or scenarios, allowing the easy comparison and analysis.

In addition, matrices such as confusion matrices will be employed to provide a visual representation of the models' classification results. These matrices showcase the number of true positives, true negatives, false positives, and false negatives, enabling a deeper understanding of the models' performance and potential areas for improvement.

By utilizing these various visual representations, we aim to provide a clear and comprehensive overview of the results obtained for each model. This approach enables a detailed analysis and facilitates the identification of strengths, weaknesses, and potential areas for optimization.

IV.6.3.1 First experiment: performance of CNN

After applying a convolutional neural network (CNN), as described in section III.2.1, and training it for a total of 30 epochs, we achieved the following results, which are presented below:

Model	**CNN**
Accuracy train	0.9983
Loss train	0.0070
Accuracy test	0.9779

Loss test	0.1026

Table 4 : *Accuracy of CNN.*

In Table 5, the results of the Confusion Matrix for CNNs are presented. The matrix includesvalues for precision, recall, and F1 score. Precision indicates the accuracy of positive predictions, recall measures the coverage of true positive instances, and the F1 score combines precision and recall to provide a balanced evaluation metric:

Class	precision	recall	f1-score	Support
Brain tumor	0.98	0.98	0.98	451
Healthy	0.98	0.97	0.98	409

Table 5 : *performance of CNNs.*

IV.6.3.1.1 performance of Resnet-50

The results obtained using the ResNet-50 model will be presented in the following format:

Model	Resnet-50
Accuracy train	0.9994
Loss train	0.0122
Accuracy test	0.9907
Loss test	0.0425

Table 6 : *Accuracy of Resnet-50.*

In table 7 results obtained using the ResNet-50 model will be presented with the accuracy value indicating the overall performance of the model.   The F1 score, which combines precision and recall, will be given for each class to provide a balanced evaluation metric.

Class	Precision	Recall	F1-Score	Support
Brain tumor	0.99	0.99	0.99	451
Healthy	0.99	0.99	0.99	409

Table 7: *performance of Resnet-50.*

IV.6.3.1.2 performance of VGG-16

Upon implementing the VGG-16 model, we achieved highly favourable and satisfactory results.

Model	VGG-16
Accuracy train	1.0000
Loss train	4.6426e-06
Accuracy test	0.9895
Loss test	0.1256

Table 8 : *Accuracy of VGG-16.*

In table 9 results obtained using the VGG-16 model will be presented with the accuracy value indicating the overall performance of the model. The F1 score, which combines precision and recall, will be given for each class to provide a balanced evaluation metric.

Class	Precision	recall	F1-Score	Support
Brain tumor	0.99	0.98	0.99	451
Healthy	0.98	0.99	0.99	409

Table 9 : *performance of VGG-16.*

IV.6.3.1.3 performance of Densnet-121

After applying the DenseNet-121 model, we obtained the following results:

Model	Densnet-121
Accuracy train	0.9887
Loss train	0.0314
Accuracy test	0.9560
Loss test	0.5147

Table 10 : *Accuracyof Densnet-121.*

In table11 results obtained using the Densnet-121 model will be presented with the accuracy value indicating the overall performance of the model.the F1 score, which combines precision and recall, will be given for each class to provide a balanced evaluation metric.

Class	Precision	Recall	F1-Score	Support
Brain tumor	0.99	0.88	0.94	451
Healthy	0.89	0.99	0.94	409

Table 11 : *performance ofDensnet-121.*

IV.6.3.2 Second experiment: performance of Convolution AutoEncoder

Table 12 displays the outcomes of the AutoEncoder model, revealing a remarkable accuracy rate of 93%:

Class	Precision	Recall	F1-Score	Support
Brain tumor	0.92	0.94	0.93	451
Healthy	0.94	0.91	0.93	409
Accuracy			0.93	860

Table 12*: performance of ConvolutionAutoEncoder.*

IV.6.3.3 Confusion matrics

In this section, we will present the confusion matrices corresponding to the experimental cases and evaluate each model to assess the obtained outcomes.

Confusion matrices provide a detailed breakdown of the predictions made by the models and their alignment with the actual ground truth. By analyzing these matrices, we can gain insights into the models' performance in terms of correctly identifying true positives, true negatives, false positives, and false negatives.

By meticulously examining the confusion matrices and analyzing the performance matrics, we can form a comprehensive understanding of the outcomes achieved by each model. This evaluation will help us draw conclusions regarding the effectiveness and reliability of the models and identify any areas that may require further improvement or optimization.

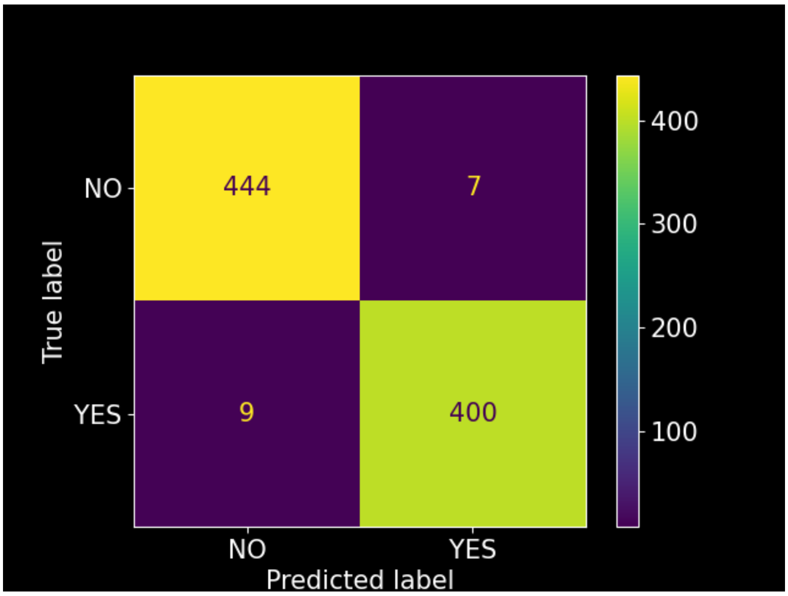


Figure 21 : *Confusion Matrix of CNNs.*

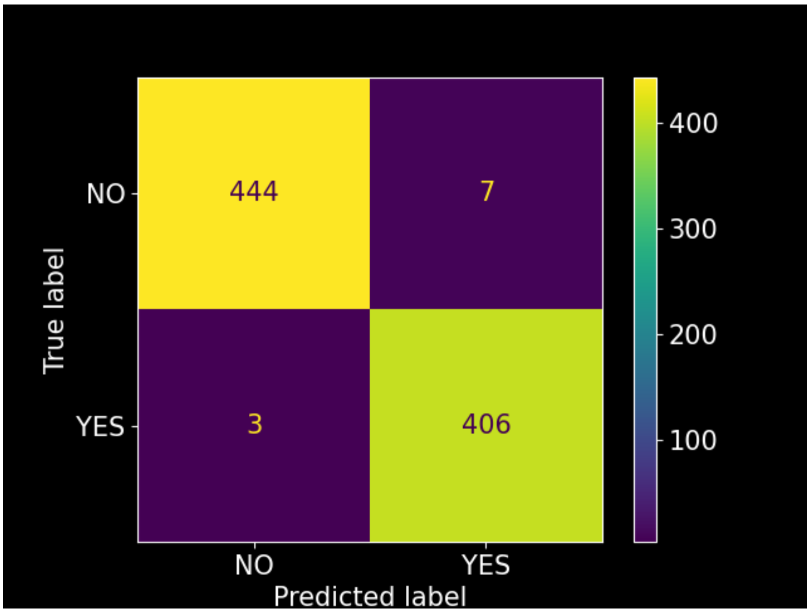


***Figure 22** : Confusion Matrix of VGG-16.*

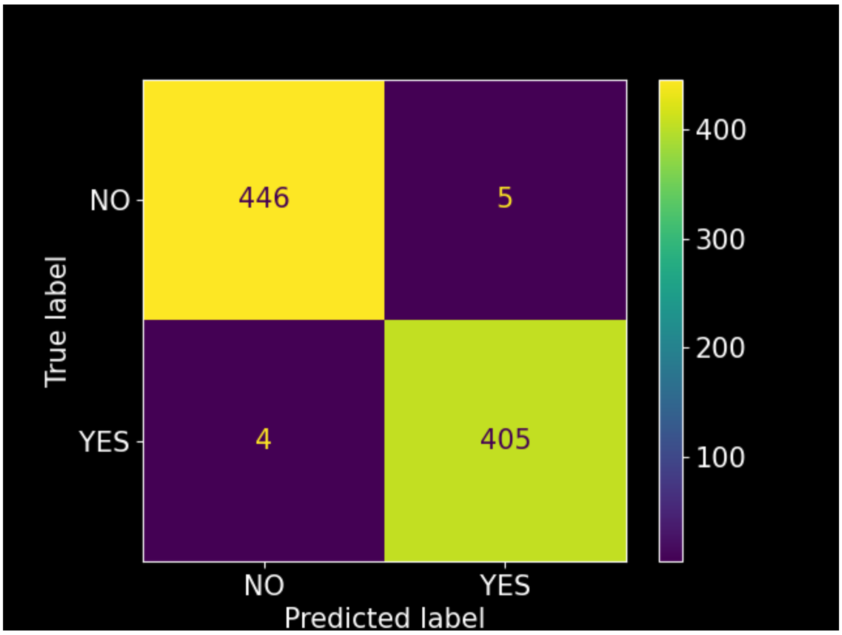


***Figure 23 :** Confusion Matrix of Resnet-50.*

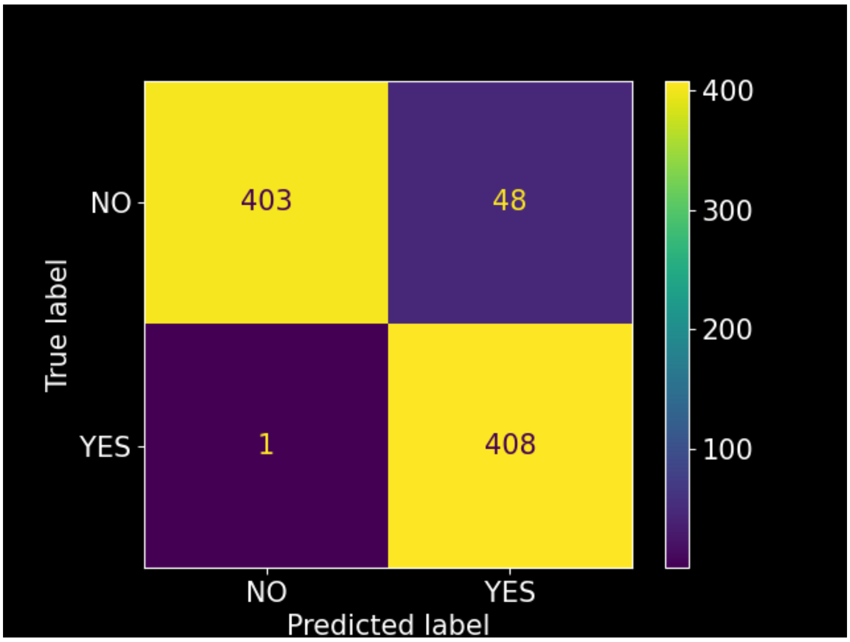


Figure 24 : *Confusion Matrix of Densnet-121.*

IV.6.3 .4The Curves

We have generated figures that depict the variations of loss with respect to epochs for both the test and validation datasets. These curves provide valuable insights into the training process and can help us understand how the models are learning and improving over time.

For each model, we have plotted separate figures showcasing the loss curves for the test and training datasets. By examining these figures, we can observe the trend of the loss values as the models undergo training. The plotted figures provide a visual representation of the loss variations, allowing us to assess the training progress and the overall performance of the models.

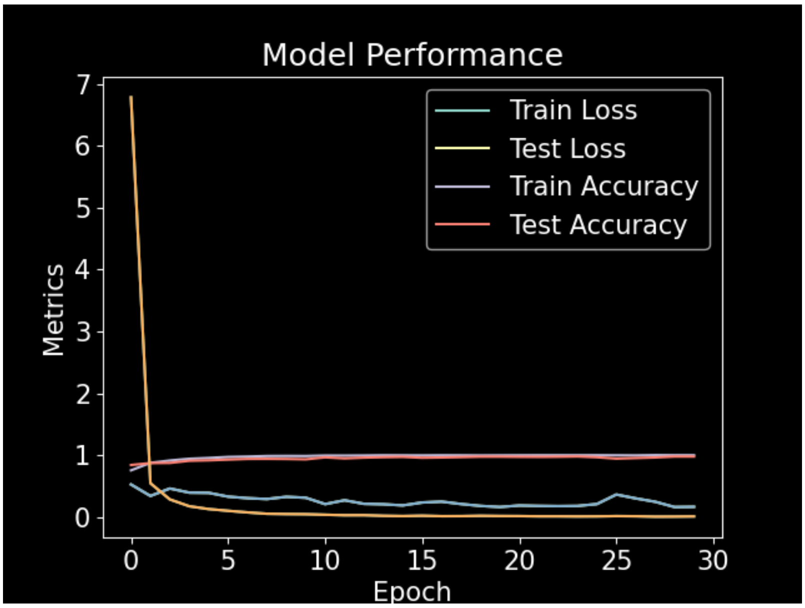


Figure 25 : *Curve of loss and accuracy the model CNNs.*

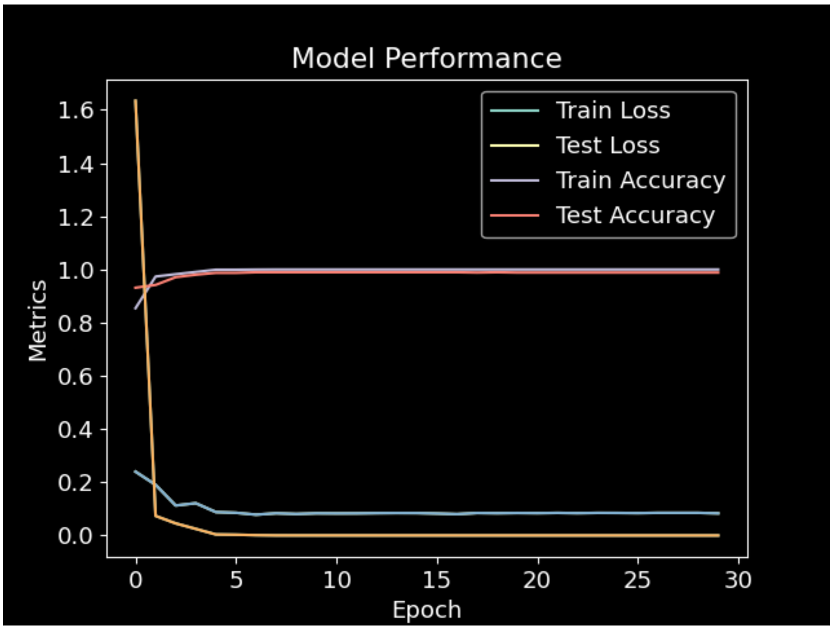


Figure 26 : *Curve of loss and accuracy the model VGG-16.*

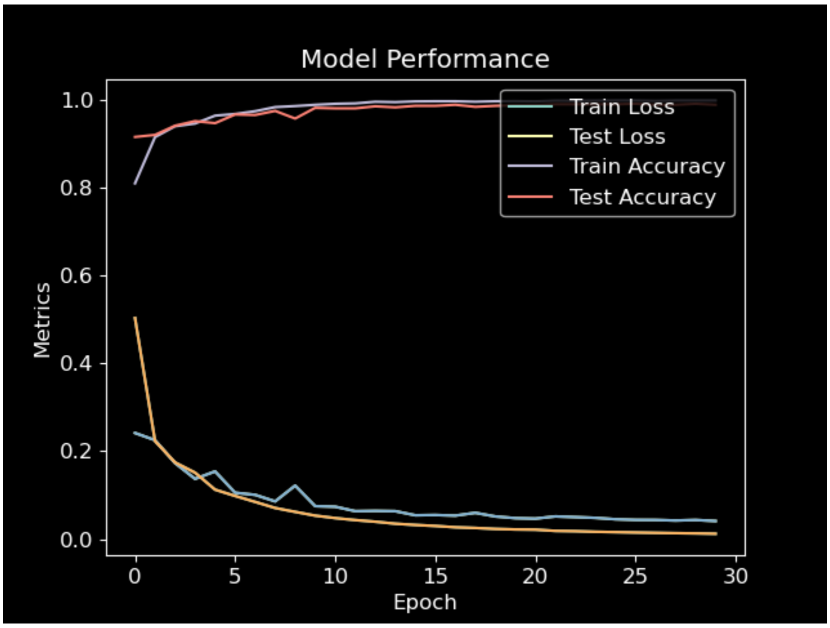


Figure 27 : *Curve of loss and accuracy the model Resnet-50.*

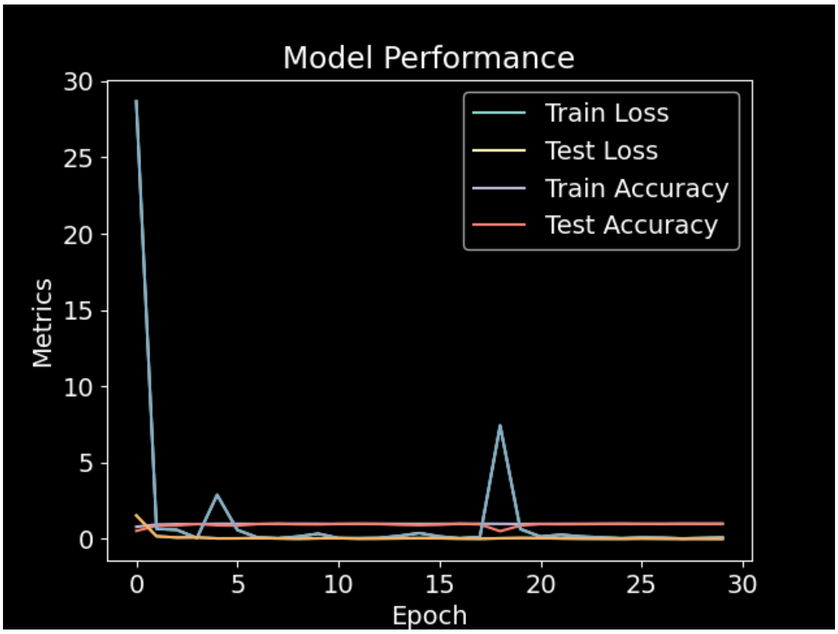


Figure 28 : *Curve of loss and accuracy the model Densnet-121.*

IV.6.3.5 Third experiment: performance of fusion method

In table 13, we will present the overall results the fusion technique and accuracy of each method,we using feature of models Resnet-50,VGG-16 and Densnet-121 training with SVM, best accuracy method is score level fusion and feature level fusion by 97%.

Methods	Accuracy
Resnet-50+VGG-16+Densnet-121 with score-level fusion	0.977
Resnet-50+VGG-16+Densnet-121 with decision-level fusion	0.955
Resnet-50+VGG-16+Densnet-121 with feature-level fusion	0.977

Table 13: *performance of fusion method.*

IV.6.3.6 Fourth experiment: performance of Resnet-50+CAE

The table below presents the outcomes obtained from merging the features of ResNet-50 and CAE using SVM and the fusion technique. Where the best result was in decision-level fusion with 98%.

Method	Accuracy
Resnet-50+CEA with score-level fusion	0.965
Resnet-50+CEA with decision-level fusion	0.989
Resnet-50+CEA with feature -level fusion	0.977

Table 14 : *performance of Resnet-50+CAE..*

IV.7 Results Discussion

After applying the previous models, we obtained the following accuracies: CNNs achieved 97%, ResNet-50 achieved 99%, VGG-16 reached 98%, and DenseNet-121 achieved 95%. When we used the fusion technique, we achieved 97% accuracy in the score-level fusion, 95% in the decision-level fusion, and 97% in the feature-level fusion. Additionally, by combining ResNet-50 with CAE using the fusion technique, we achieved 96% accuracy in the score-level fusion, 98% in the decision-level fusion, and 97% in the feature-level fusion. we applied the fusion technique to assess if we could improve or worsen the results, and although the outcomes varied, they were deemed highly acceptable.

IV.8 Conclusion

In this chapter, we provided a comprehensive overview of our study. We began by describing the dataset we used, highlighting its relevance and importance to our research. We then detailed the materials and resources we utilized throughout our study, including the models and fusion techniques.

Next, we presented the experimental results obtained from our two architectural approaches. In the first approach, we combined ResNet-50, VGG-16, DenseNet-121, and employed the fusion method. We carefully analyzed and evaluated the performance of this approach, considering various metrics such as accuracy, precision, recall, and F1-score.

In the second approach, we focused on the fusion of ResNet-50 and CEA models, utilizing the fusion method. We conducted experiments and analyzed the results to assess the performance of this approach. We compared and contrasted the outcomes with the previous approach and identified any improvements or differences achieved through the fusion of these models.

At the end, we are delighted to announce that our study has yielded highly satisfactory outcomes.

General conclusion

Deep learning has gained significant popularity in recent years among researchers and big enterprises. It is an important component of machine learning and artificial intelligence systems. Deep learning finds applications in various fields such as remote sensing, healthcare, and intelligent transportation.

One of the key reasons for the popularity of deep learning is its ability to simulate operations of the human brain using computers. Tasks like knowledge acquisition, learning, reasoning, recognition, and classification can be effectively performed through deep learning algorithms. This resemblance to human brain functioning has opened up new possibilities for solving complex problems and enhancing AI systems.

This thesis focuses on the development of deep learning models for brain tumor detection in MRI images. The research process involved several steps. Firstly, we curated and organized a dataset, which was then divided into a training set and a testing set. Next, we employed a Convolutional Neural Network (CNN) model to analyze the images.

To enhance the analysis further, we leveraged pre-trained models such as VGG-16, ResNet-50, and DenseNet-121 to extract essential features from the dataset. These extracted features were then classified using a Support Vector Machine (SVM) with feature fusion techniques, which helped in improving the accuracy of tumor detection.

Additionally, we explored the use of the ResNet-50 model in combination with a self-attentive encoder called CEA for classification. This approach involved extracting features from the ResNet-50 model and using them along with the CEA and SVM with fusion techniques to classify the MRI images.

By combining deep learning techniques, pre-trained models, and feature fusion methods, we aimed to develop a robust and accurate brain tumor detection system for MRI images. The utilization of these methodologies allowed us to achieve improved performance and enhance the overall accuracy of the classification process.

Reference:

[1] « ARTIFICIAL INTELLIGENCEr.pdf ».
[2] J. Hurwitz, « Machine Learning For Dummies®, IBM Limited Edition », 2018.
[3] O. G. Yalçın, « 4 Machine Learning Approaches that Every Data Scientist Should Know », *Medium*, 2 février 2021. https://towardsdatascience.com/4-machine-learning-approaches-that-every-data-scientist-should-know-e3a9350ec0b9 (consulté le 3 juin 2023).
[4] « Deep Learning by Ian Goodfellow, Yoshua Bengio, Aaron Courville (z-lib.org).pdf ».
[5] V. Tyagi, *Understanding Digital Image Processing*. 2018. doi: 10.1201/9781315123905.
[6] « Full Text PDF ». Consulté le: 12 mai 2023. [En ligne]. Disponible sur: https://www.researchgate.net/profile/Debashis-Ganguly/publication/227172822_Medical_Imaging_A_Review/links/54d952960cf24647581ddf60/Medical-Imaging-A-Review.pdf
[7] « X-Rays ». https://medlineplus.gov/xrays.html (consulté le 14 mai 2023).
[8] C. for D. and R. Health, « What is Computed Tomography? », *FDA*, sept. 2020, Consulté le: 14 mai 2023. [En ligne]. Disponible sur: https://www.fda.gov/radiation-emitting-products/medical-x-ray-imaging/what-computed-tomography
[9] https://www.facebook.com/verywell, « What You Should Know About MRI », *Verywell Health*. https://www.verywellhealth.com/what-is-an-mri-and-what-does-it-do-3157069 (consulté le 5 juin 2023).
[10] « Magnetic resonance imaging of the brain », *Wikipedia*. 7 mai 2023. Consulté le: 14 mai 2023. [En ligne]. Disponible sur: https://en.wikipedia.org/w/index.php?title=Magnetic_resonance_imaging_of_the_brain&oldid=1153601287
[11] « introduction-to-artificial-neural-networks.pdf ».
[12] E. Todt et B. A. Krinski, « Convolutional Neural Network - CNN ».
[13] « s40537-021-00444-8.pdf ».
[14] « CNN_Presentation.pdf ».
[15] « Figure 3. Depiction of the convolution layer with a filter in... », *ResearchGate*. https://www.researchgate.net/figure/Depiction-of-the-convolution-layer-with-a-filter-in-convolutional-neural-network-CNN_fig3_338190342 (consulté le 30 avril 2023).
[16] Prabhu, « Understanding of Convolutional Neural Network (CNN) — Deep Learning », *Medium*, 21 novembre 2019. https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148 (consulté le 3 juin 2023).
[17] « Fig. 4. Pooling layer operation oproaches 1) Pooling layers: For the... », *ResearchGate*. https://www.researchgate.net/figure/Pooling-layer-operation-oproaches-1-Pooling-layers-For-the-function-of-decreasing-the_fig4_340812216 (consulté le 30 avril 2023).
[18] « What are Convolutional Neural Networks? | IBM ». https://www.ibm.com/topics/convolutional-neural-networks (consulté le 5 juin 2023).
[19] « Maggipinto et al. - 2018 - A Convolutional Autoencoder Approach for Feature E.pdf ».

[20] « Convolutional Autoencoders », *Paperspace Blog*, 14 octobre 2022. https://blog.paperspace.com/convolutional-autoencoder/ (consulté le 23 mai 2023).
[21] N. Tomar, « Building Convolutional Autoencoder using TensorFlow 2.0 », *Idiot Developer*, 20 mai 2020. https://idiotdeveloper.com/building-convolutional-autoencoder-using-tensorflow-2/ (consulté le 23 mai 2023).
[22] « ReLU function », *AILEPHANT*. https://ailephant.com/glossary/relu-function/ (consulté le 30 avril 2023).
[23] « Oppermann - Loss Functions in Deep Learning.pdf ».
[24] K. He, X. Zhang, S. Ren, et J. Sun, « Deep Residual Learning for Image Recognition ». arXiv, 10 décembre 2015. Consulté le: 3 mai 2023. [En ligne]. Disponible sur: http://arxiv.org/abs/1512.03385
[25] « He et al. - 2016 - Deep Residual Learning for Image Recognition.pdf ». Consulté le: 9 juin 2023. [En ligne]. Disponible sur: https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf
[26] K. Simonyan et A. Zisserman, « Very Deep Convolutional Networks for Large-Scale Image Recognition ». arXiv, 10 avril 2015. Consulté le: 3 mai 2023. [En ligne]. Disponible sur: http://arxiv.org/abs/1409.1556
[27] « Huang et al. - 2017 - Densely Connected Convolutional Networks.pdf ».
[28] « jakkula.pdf ».
[29] « support-vector-machinessvm-a-complete-guide-for-beginners (1).pdf ».