University Of Kasdi Merbah Ouargla

Faculty Of Information Technology And Communication Department Of Information Technology

Dissertation

Domain: Mathematics and Computer Sciences

Field: Information Technology

Specialty: Fundamental Computer Sciences

***Presented by**:* Oualid Bennahia and Dia Elhak Temmar

Covid-19 detection from chest radiography (CXR) images using deep learning

Defended publicly At: 19/ 06 / 2022

The jury:

Dr. BEKKARI Fouad	**President**	**UKM Ouargla**
Dr. AIADI Ouassama	**Supervisor/reporter**	**UKM Ouargla**
Dr. KHALDI Bilal	**Examiner**	**UKM Ouargla**

Academic year: 2021 /2022

Table of Contents

Table of Figures

List of Tables

Acknowledgment

First and foremost, praise and thanks to ALLAH the sacred and the mighty for giving me the ability and helping me in accomplishing this thesis.

We would like to express our immense gratitude to our supervisor Dr. AIADI Oussama, for his advice, support, guidance and inspiration to conduct our research.

I gratefully acknowledge all the professors within the Department of computer sciences and information technology.

Dedication

See my graduation to my dear Father, may Allah rest his mercy.

To my mother

To my sisters and brothers and my nephew Djad

May Allah protects them

To my friends
To all who supports me

Dia elhak temmar

To my mother and my Father may Allah protects them

To my sisters and brothers

To my friends
To all who supports me

Oualid Bennaha

Abstract

The wide spread of the coronavirus since the beginning of 2020 has created a global crisis, causing the death of a large number of people, the total or partial closure of international borders, the stagnation of the global economy followed by the loss of many people's jobs, overcrowding in hospitals and the disability of medical staff, as a result, a great deal of research has been carried out to combat the coronavirus, both medically and in terms of AI, especially deep learning, in this thesis, we are interested in developing models that can detect Covid-19 from chest radiography (CXR) images using deep learning.

The proposed approach is comprised of four stages. First, which is preparing the dataset we used by dividing it into training group and testing group, second, extracting features of our dataset images using our models Resnet-50 and convolutional auto encoder(CAE), one of the advantages of using CAE and Resnet-50 it proves they are complementary between supervised and unsupervised deep architectures, third, classify the features that we obtained from the previous step using Support vector machine images, finally, improving the SVMs result by using fusion technique like max, min, product and sum and mean .

Our dataset contains three classes CRIs images for Normal people and people with Covid-19 and Viral Pneumonia with 12123 images in total 9091 for training and 3032 for testing, experimental results proved the effectiveness of the proposed method. The proposed ensemble CNNs has outperformed several relevant states of the art method, to clarify, we get a 98.74% accuracy as a result of fusion (score level fusion (sum, product, mean) four SVM, in end we build site web using our method.

Keywords: *Covid-19, Deep learning, CNN, Medical image, fusion technique, CXR.*

ملخص

أدى الانتشار الواسع لفيروس كورونا منذ بداية عام 2020 إلى حدوث أزمة عالمية، التسبب في وفاة عدد كبير من الناس، والإغلاق الكلي أو الجزئي للحدود الدولية، وركود الاقتصاد العالمي الذي أعقبه فقدان وظائف كثير من الناس، الاكتظاظ في المستشفيات وإعاقة الطاقم الطبي نتيجة لذلك، تم إجراء قدر كبير من الأبحاث لمكافحة فيروس كورونا، سواء من الناحية الطبية أو من حيث الذكاء الاصطناعي، وخاصة التعلم العميق، في هذه الأطروحة، نحن مهتمون بتطوير نماذج يمكنها اكتشاف Covid-19 من صور التصوير الشعاعي للصدر (CXR) باستخدام التعلم العميق.

ويتألف النهج المقترح من أربع مراحل. أولاً، نقوم بإعداد مجموعة البيانات المستخدمة عن طريق تقسيمها إلى مجموعة تدريب و مجموعة اختبار، ثانيًا، استخراج ميزات صور من مجموعة البيانات الخاصة بنا باستخدام نماذجنا Resnet-50 ومشفر ذاتي تلافيفي(CAE) ، إحدى مزايا استخدام CAEو Resnet-50 هي إثبات وجود تكامل بين البنى العميقة الخاضعة للإشراف وغير الخاضعة للإشراف ، ثالثًا، تصنيف الميزات التي حصلنا عليها من الخطوة السابقة باستخدام صور آلة ناقل الدعم، أخيرًا، يؤدي تحسين SVMs باستخدام تقنية الاندماج مثل الحد الأقصى و الحد الأدنى والجمع والضرب والمتوسط.

تحتوي مجموعة البيانات الخاصة بنا على ثلاث فئات من صور CRIs للالتهاب الرئوي الطبيعي people,Covid-19 والفيروسي مع 12123 صورة في المجموع 9091 للتدريب و 3032 للاختبار، أثبتت النتائج التجريبية فعالية الطريقة المقترحة. تفوقت المجموعة المقترحة من CNNs على العديد من الحالات ذات الصلة بالطريقة الفنية، للتوضيح، نحصل على دقة 98.74٪ نتيجة الاندماج (اندماج مستوى النتيجة (الجمع والضرب والمتوسط)) أربعة في إجمالي SVMs.

تحتوي مجموعة البيانات الخاصة بنا على ثلاث فئات من صور CRIs للأشخاص العاديين و للأشخاص مصابين Covid-19 والالتهاب الرئوي الفيروسي مع 12123 صورة في المجموع 9091 للتدريب و 3032 للاختبار، أثبتت النتائج التجريبية فعالية الطريقة المقترحة. تفوقت المجموعة المقترحة من CNNs على العديد من الحالات ذات الصلة بالطريقة الفنية، للتوضيح، نحصل على دقة 98.74٪ نتيجة الاندماج (اندماج مستوى النتيجة (الجمع والضرب والمتوسط)) أربعة SVMs، وفي النهاية نبني الموقع على شبكة الإنترنت باستخدام طريقتنا.

الكلمات المفتاحية: *Covid-19، التعلم العميق ، CNN، الصورة الطبية، تقنية الاندماج، CXR.*

‘

Chapter I

General Introduction

1.1 Introduction:

The coronavirus pandemic has spread widely since the end of 2019 [1], with more than 532,201,219 confirmed cases and 6,305,358 deaths as of June 10, 2022 [2].
There are tests for Covid-19 was discovered that can detect either SARS-CoV-2, the virus that causes COVID-19, or antibodies that your body makes after getting COVID-19 or after getting vaccinated. Moreover, the first type can take days to complete and include RT-PCR and other types of NAATs in Laboratory tests [3].

In the same way, many attempts have been made to use deep learning to detect COVID-19 infection; in general, deep learning (DL) is one of the branches of artificial intelligence that has seen exponential growth in recent years. The scientific community has focused its attention on DL due to its versatility, high performance, and high generalization capacity [4].

In addition, Deep learning is a subset of machine learning, which is mostly uses a neural network with three or more layers, these neural networks attempt to simulate the behavior of the human brain albeit far from matching its ability to allow it to “learn” from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy. [5]

Deep Learning is extensively used in the medical field, particularly in areas where the diagnostic information a doctor examines is already digitized, such as:

- Detecting lung cancer or strokes based on CT scans [6].
- Assessing the risk of sudden cardiac death or other heart diseases based on electrocardiograms and cardiac MRI images [7].
- Classifying skin lesions in skin images [8].
- Finding indicators of diabetic retinopathy in eye images [9].

1.2 Problematic:

In this thesis, we consider handling the research question concerned with the classification of medical images (CRI), which originally belonged to three categories, images of healthy people, people with covid-19, and people with Viral Pneumonia. Several deep learning models were built that showed very good results in classifying images from previous categories, such as CNN and RNN or SVM as well as integrating results from more than one technology. The most difficult challenge to such a problem is, accuracy must be so high as to be almost ideal because any misclassification of one person out of ten may cause disease transmission to more people.

1.3 Overview on the Related Work :

Since the beginning of the spread of covid-19, many researchers proposed different techniques for detecting the virus from CRIs. Most research focused on DL techniques to detect the COVID-19 from CRI of patients, some of them are interested in COVID detection from non-COVID cases. Others were concerned with a three-class classification (COVID versus Normal versus pneumonia), and the others focused on COVID-19 detection using a four-class classification (COVID vs Normal vs Bacterial pneumonia vs Viral pneumonia).

Ahmed S. Elkorany and Zeinab F. Elsharkawy [10] built a DL model called it COVIDetection-Net that has focused on extracting features from dataset images using two models: ShuffleNet and SqueezeNet and then passing these features to MSVM to train for classification, this gives a 94.44% accuracy for 4-classes and 99.72% for 3-classes and 100% for 2-classes.

Sohaib Asif , Kamran Amjad built a model COV-19Ne with three convolution layers and two 2D Max Pooling layers, and two fully connected layers, this model gives a 99.45% accuracy for binary classification[11].

Dipayan Das et al [12] presented truncated inception net to detect COVID-19 positive from non-COVID cases using CRIs. Detection accuracy of 99.9 % is obtained.

Sethy et al. [13] extracted the deep features from the CRIs using the fully connected layer of the pre-trained models. Then, features matching have been done using SVM classifier, the authors utilized 13 different pre-trained net and the highest classification accuracy of 98.66 % was obtained using ResNet50.

1.4 Motivations:

On the other hand, many AI researchers, especially deep learning, have also started to build models that contribute to the discovery of viruses, this has allowed many different models to be built in terms of structure and has produced different and good results, The purpose of all this was to develop the scientific research aspect in the field of artificial intelligence and specifically deep learning in addition to that we cannot negate the likelihood of the emergence of a virus or similar disease so that such models provide a significant contribution from the face of such potential disasters, on the other hand, these models can be circulated to other diseases that already exist. There are also models built to detect diabetes or cancer.

Many models were built to detect coronavirus, but the use of unsupervised learning as a convolutional auto encoder was rare [14] unlike the CNNs Models as we mention in the Overview on the Related Work section.

The reasons why we chose this topic is because we wanted to build our model and thus exploit what we learned in previous years and hire it to solve the pandemics that caused a major collapse of several fields globally, especially nationally.

1.5 Contributions:

The contributions of this thesis can be summarized as follows:

- Adding another work for COVID 19 detection that uses unsupervised learning (CAE)
- Use a pre-trained model(resnet50) and CAE in extracting features from CRIs rather than classifying them directly
- Uses a support vector machine to classify CRIs based on extracted features of each previous resnet50 and CAE.
- Fusion of the results obtained from each SVM to increase the accuracy of the prediction

1.6 Thesis structure:

This thesis is organized as follows:

- A states the general background of the work involving definition of Digital image and deep learning ,applications of deep learning, classification methods.

- The reports experimental results after conducting experiments on the dataset after classification using SVM on the features was extracted by resnet-50 model and CAE.

-A general conclusion is given with some perspectives that can be useful.

Chapter II

Work background

2.1 Introduction:

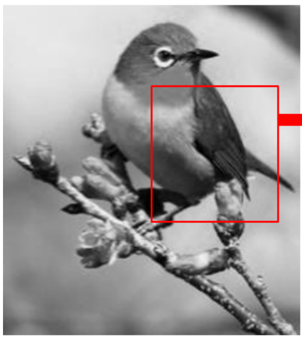
In this chapter, we are interested in presenting the main background of our work, including details about digital image processing, deep learning, types of deep learning algorithms, applications of deep learning, components of recognition systems, classification methods and lastly an overview on features extraction ,convolutional neural network (CNN),convolutional auto encoder (CAE),support vector machine (SVM).

2.2 Digital image:

By referring to Gonzelez and Woods book [15], we can quote the following definition for the digital image:

"An image may be defined as a two-dimensional function, *f(x, y)*, where x and y are spatial (plane) coordinates, and the amplitude of *f* at any pair of coordinates *(x, y)* is called the intensity or gray level of the image at that point. When *x*, *y* and the amplitude values of *f* are all finite, discrete quantities, we call the image a digital image".

Based on this definition, the authors have highlighted that image is similar to a discreet 2D function, where the pixel values represent the function image and the pixel coordinates (x, y) represent the 2D inputs of that function. as follows (Fig. 1):



$f(x,y)$

y

x

54	45	59	65	45	22	12	150	125	201
0	254	88	112	21	78	5	78	102	45
78	77	54	96	120	55	22	99	120	204
54	8	52	78	45	45	12	68	183	7
78	45	23	125	78	35	58	25	120	5
58	95	241	55	78	45	445	100	45	0
33	11	20	57	10	5	8	0	55	0
147	52	255	125	42	15	89	6	87	88

Figure : Digital image

2.3 Digital image processing:

Digital Image Processing is the manipulation and analysis of the digital data with help of computer algorithms and efficient algorithms to output hidden information, it helps also in enhancing image quality and removing noise and detecting edges.

2.4 Deep learning:

By referring to Ian Goodfellow and Yoshua Bengio, Aaron Courville [16], we can quote the following definition for the Deep learning:

“Deep learning is a form of machine learning that enables computers to learn from experience and understand the world in terms of a hierarchy of concepts. Because the computer gathers knowledge from experience, there is no need for a human computer operator to formally specify all the knowledge that the computer needs. The hierarchy of concepts allows the computer to learn complicated concepts by building them out of simpler ones; a graph of these hierarchies would be many layers deep”.

2.5 Applications of deep learning:

Deep learning has various applications in many areas of artificial intelligence (AI). In the following, a brief summary of those applications is presented:

- Image classification is the process of automating the process of categorizing an image. Many applications take images as input and return classification for these images (e.g., "cat" for a picture of a cat, "person" for a picture of an individual standing). There are many difficulties in teaching computers to distinguish between classes by hand-crafting rule sets and manually training algorithms that use those rules.

-Speech recognition can be defined as the ability to understand the spoken words of the person speaking.

- Question answering: computers can answer factual questions about a specific topic. Artificial intelligence is already assisting customer service representatives at many companies like AT&T and Google.

- Handwriting recognition: By reading to Miika Silfverberg [17], we came up with the following definition for the Handwriting recognition:

Handwriting recognition is the ability to identify written from images of Handwriting and all this process be by computer or device, it utilises input images as format that the computer understands (e.g., Unicode text).

- Emotion detection: some are working on a program that can recognize the emotions of an individual through facial analysis. The company Affectiva, is already creating applications for technology giants like Apple and Intel. In 2012, Intel purchased Affectiva for $50 million (£32 million).

- Translation: a computer translates from one language to another. The most famous example is Google Translate. All of these applications are made possible due to deep learning algorithms.

2.6 Features extraction :

Features are representative numerical values that are designed to reflect the image content, features extraction is extracting important values from the image and it involves reducing the number of resources required to describe a large set of data. For example, reconstruction data and pattern recognition…etc. Many Deep learning practitioners depend on properly optimized feature extraction as the key to effective model construction, also in features learning is done automatically in deep learning in contrary to handcrafted features.In machine learning, pattern recognition, and image processing, feature extraction refers to the process of transforming primary data into digital features that can be processed while preserving the information in the original data set (Fig. 2).

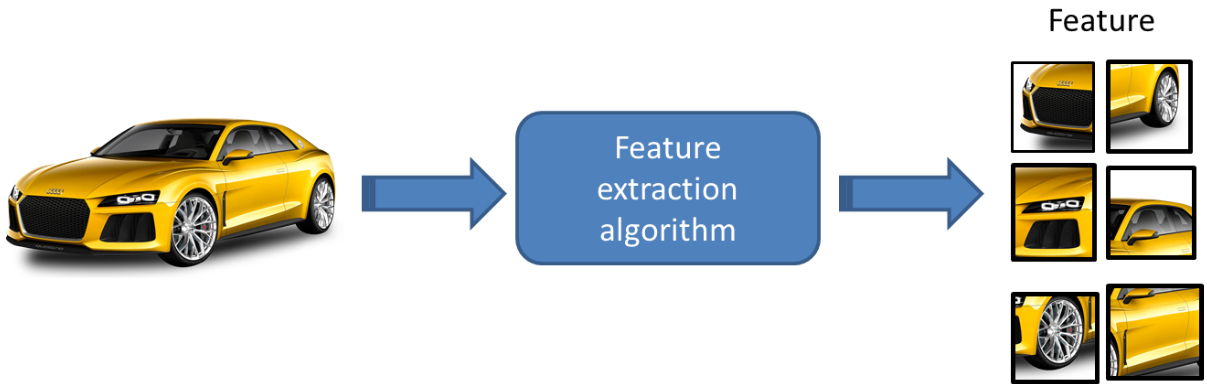


Figure : Feature Extraction

In the following, we present an overview of certain image features of different categories hog, lbp, colour and histogram.

Histogram of Oriented Gradients (hog):

The histogram of oriented gradients is a feature descriptor, also known as HOG, it is used in computer vision and image processing. The method of HOG is similar to edge orientation histograms and shape contexts, SIFT. The histogram of oriented gradients descriptor concentrates on the shape of an entity , it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy as follows (Fig. 3):

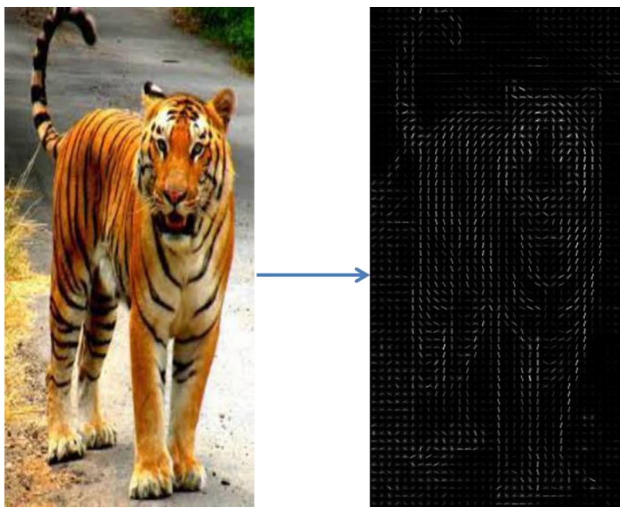


Figure the method of HOG

Local binary patterns (LBP):

Feature LBP is a simple method form, but very efficient and widely used like classification in images and texts face recognition, fingerprint identification. In LBP, it is shown in the following Figure (Fig. 4) .Frist, it works by taking part of the image and generate LBP code by threshold the values in a neighbourhood with old threshold placing. Second, multiplying the resulting binary map with a predefined mask. Third, sum the values to obtain an 8-bit LBP[18].

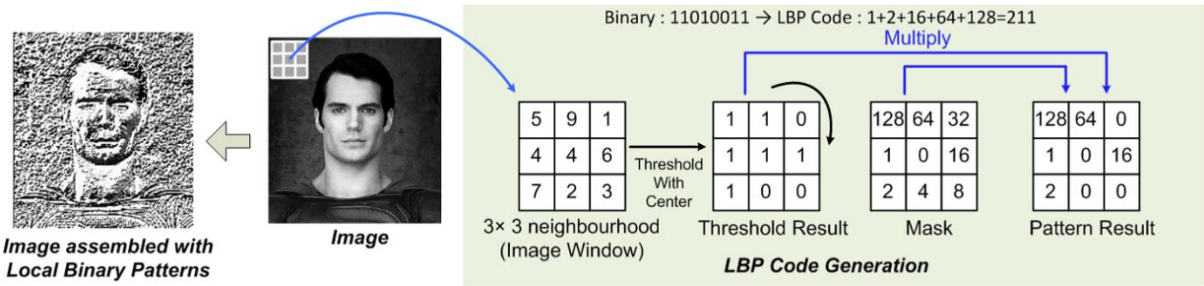


Figure : the process Local binary patterns [18]

Colour feature:

Colour features are designed to encode colour information in image, it is one of the most important descriptors. In colour representation, there are various spaces used for three-dimensional spaces like RGB or HSV. Examples on colour features include colour histogram (Fig. 5), colour moments dominant colours…etc.

Figure : colour histogram for three different images

In the Most algorithms used in feature extraction, we can found autoencoders and principle components analysis (PCA) and independent component analysis (ICA), linear discriminant analysis (LDA)…etc.

2.7 Classification methods :

Classification in machine learning is one of supervised learning categories. In the terminology of deep learning, the term ‘classification’ identifies the category of new observations based on training data. In the model, a program learns from the given dataset or observations and then classifies new observations into some classes or groups (i.e., yes or no, spam or not spam, cat or dog). Classes can be called targets/labels or categories. Types of classification can be of three types: binary classification, multiclass classification. in supervised classification algorithms, they exist many examples for instance support vector machine (SVM) and convolutional neural network (CNN), also convolutional autoencoder (CAE) are unsupervised learning methods but it used in supervised learning.

In the following, we will shed a light on the convolutional neural network (CNN) and convolutional autoencoder (CAE), support vector machine (SVM).

2.8 Convolutional Neural Network (CNN) :

Convolutional Neural Networks (CNNs), also known as (ConvNets), is one of artificial neural network (ANN) categories. ConvNets is a method of deep learning that is mainly used in image classification and detection, segmentation, it also based on multiple layers, it is used for image processing and objects detection.The ConvNets is based on a number of sequential processes: the process of extracting characteristics through the layers process: where we do a filter that scans the image step by step and extracts the distinctive elements in it, that step can be repeated more than once and every time we change the size of the filter.

2.8.1 History of Convolutional Neural Network (CNN)*:*

The origin of ANN back to the first attempts tried to imitate the human perception in the 20th century (Frank Rosenblatt in 1957). Frank Rosenblatt was American psychologist notable in the field of artificial intelligence and all the basic ingredients of the deep learning systems of today ,they are developed and explored by him.The first architecture of CNNs was called Neocognitron by Kunihiko Fukushima in 1980, It was inspired by Receptive fields in the visual cortex by works of Hubel and Wiesel in 1950 and 1960. Thenceforth, many researchers have depended to understand all these Works, then automating them to further the build of Neural Networks.The first name convolutional neural networks shown by Yann LeCun, a postdoctoral computer science researcher in the 1980s and it actually originated with the design of the LeNet in 1988, it was used for character recognition tasks.Then, ImageNet started in Large Scale Visual Recognition Challenge (ILSVRC) between 2010 and 2017 and to come to the best classification performance by mode of more than 14 million images from different classes.Since then many CNN architectures were designed to join Challenge involving ResNet-152, Xception, VGG16, SuperVision/AlexNet,...etc.ConvNets has been applied for different domains not only in Image recognition, here we outline some of these applications:

- Natural language processing (NLP) .
- Video analysis .
- Anomaly Detection.
- Drug discovery.
- Health risk assessment and identification of biomarkers of aging.

- Checkers game.

2.8.2 Architecture of Convolutional Neural Network (CNN)*:*

CNN has multiple stacked layers in hidden layers that are used in extracting details from an image. The four important layers are convolution and ReLU layer, pooling layers, where after layers include fully connected. Each layer has the following description:

Convolution layer:

This layer is also known as the first layer in hidden layers and it is the central building block of ConvNets. It works as a feature sensor and it includes a collection of kernels (or filters) for detecting lower-level features such as edges, corners and flat lines, the parameters of layers have been learned from the training. Filters have the same dimension but are smaller than the actual image every filter convolves with the image to create a filter map. The kernel passes through the entire image on the top-bottom and right-left by looking at a pixel each time and doing the operation mentioned in the Fig 6:

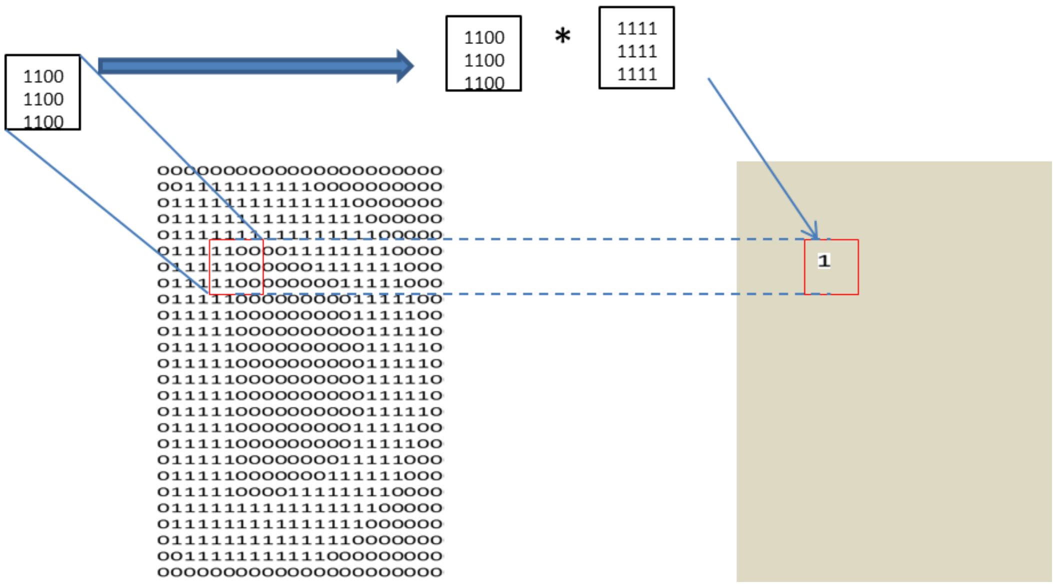


Figure : Convolution operation

In mathematics, we use in convolution operation an image I with size $N \times N$ and kernel with size $M \times M$ has the following expression:

$$conv(I,K)_{x,y} = \sum_{i=1}^{M} \sum_{j=1}^{M} K[i,j] \times I[x+i-1, y+j-1]$$

In this layer have certain parameters to adjust, which are:

Stride:

This Stride refers to the moving step (from top to bottom and from right to left) , The value of Stride controls how deep columns around the height and width are given by how many pixels to escape from the current pixel to the next pixel to be considered .

Padding:

Padding use to control the output size .The padding solve problem of information loss from edges .

Rectified Linear Units layer (ReLU):

In this layer, an element-wise function $f(x) = max(0, x)$ with input x. It will be applied to point where the negative ones are converted to zero and positive values are preserved, fig 7:

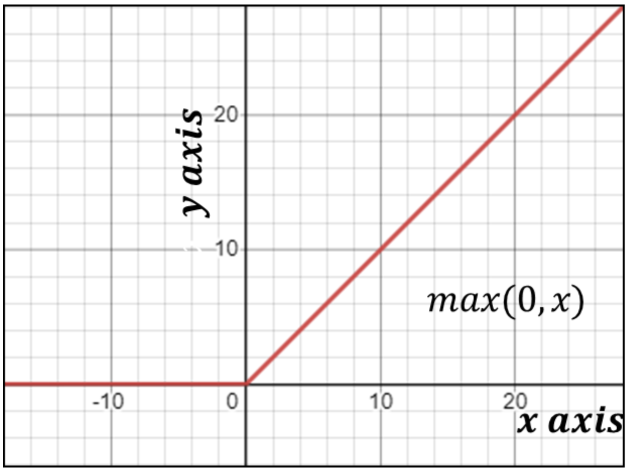


Figure : ReLU function

Pooling layers:

This layer work to reduce the size of feature maps produced by convolution layer and it is a form of non-linear down-sampling, noting that important information is kept for consideration in the next layer. There are three types of pooling, average and min, max pooling. The first one takes an average pooling considers the average of the region, while the second, the min value within a certain region, while the last (Fig. 8), the max value within a certain region(e.g., 2×2).

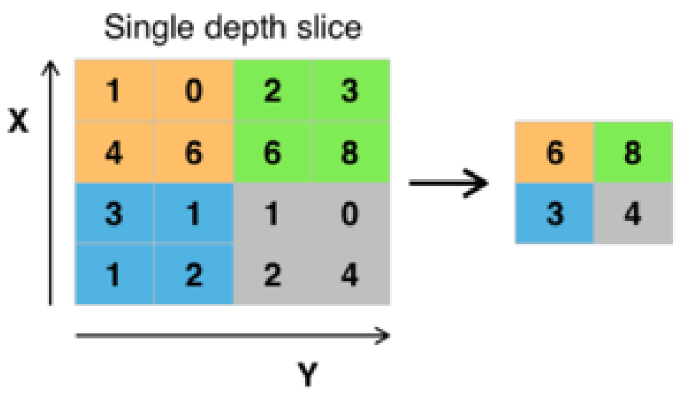


Figure : max pooling

Fully-connected Layer (FC Layer):

In the end of CNN this layer can be found it , it is called fully connected, also known as dense layer which is primarily a similar to one from a feed-forward neural network, because all neurons from the previous layer are connected to activation functions and used to flatten results prior to classification stage (Fig.9):

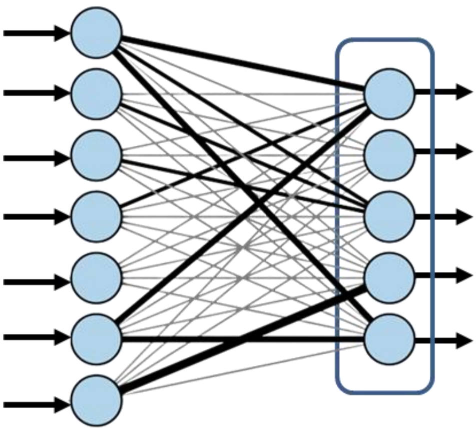


Figure : Fully-connected Layer

In the end whole CNN architecture looks as follows (Fig. 10):

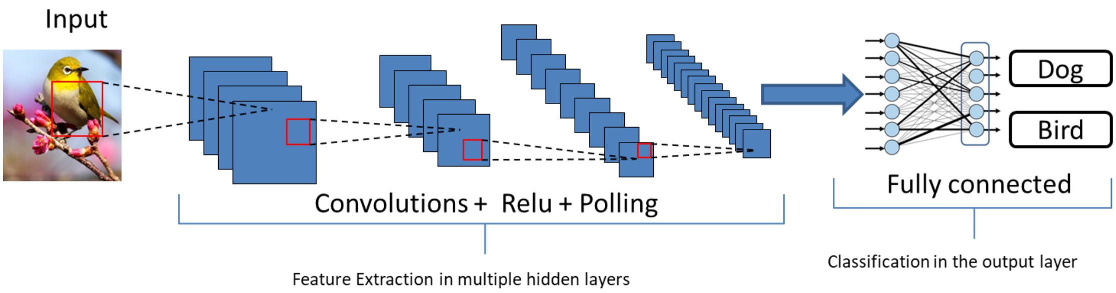


Figure : CNN architecture

Note that a back-propagation algorithm is used to optimize the CNN by updating its weights in each epoch, on the basis of its partial derivatives. Weights are changed the parameters will be gradually changed until reaching the final values.

2.9 Convolutional Auto-Encoder (CAE) :

CAE is a popular type of autoencoder; also it is a variation of neural networks which are used as unsupervised learning tools of convolution filters. Generally, they are involved in the task of image reconstruction to minimise reconstruction errors by understanding the optimal filters. CAE is used mostly in Feature Extraction (i.e classification or regression) and image denoising (i.e Image Noise Reduction), image generation, dimensionality reduction.

2.9.1 How does a convolutional autoencoder work?

This CAE teaches to encode the entry image in a group of easy weights (signals) and then rebuild the entry from them. Also, we can change the geometry or generate the reflectance of the image by using CAE. This model, it has two types of layers, encoder layers are known as convolution layers and decoder layers are also called deconvolution layers. The deconvolution side is also known as transpose convolution or upsampling .

2.9.2 The architecture of Convolutional Autoencoder *:*

In the end whole CAE architecture looks as follows (Fig. 11):

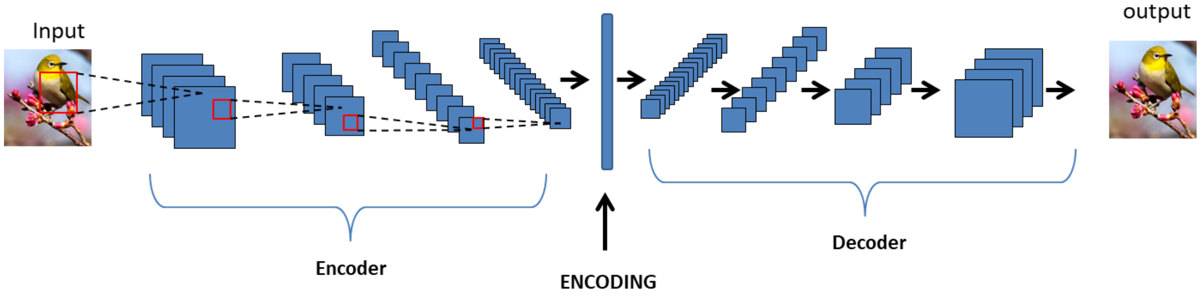


Figure Architecture of Convolutional Autoencoder

2.10 Support Vector Machine (SVM) :

In supervised machine learning algorithms , SVM is a famous model that are supervised method that sees at the input data and sorts it into one of two classes by analyses data .

SVMs are used in lot of recognition system like image classification, text categorization, handwriting recognition… etc.

A support vector machine is also known as a support vector network (SVN). [19]

2.10.1 How does a Support Vector Machine work?

In the process of SVM that are worked with map data from a high-dimensional feature space, so that information points can be classified, even when the information are not otherwise linearly separable. A separator between the categories is found, later the information is changed in such a way that the separator could be drawn as a hyper-plane.

The mathematical function used for the transformation is known as the kernel function. SVM in supports the following kernel types:

- Linear, equation is $f(x) = ax + b$
- Polynomial, equation is $k(x_i, x_j) = (x_i . x_j + 1)^d$
- Radial basis function (RBF), equation is $k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$
- Sigmoid, equation is $k(x, y) = \tanh(\alpha x^T y + c)$

In the end whole Support Vector Machine architecture of each kernel (linear, RBF, Polynomial) looks as follows (Fig. 12):

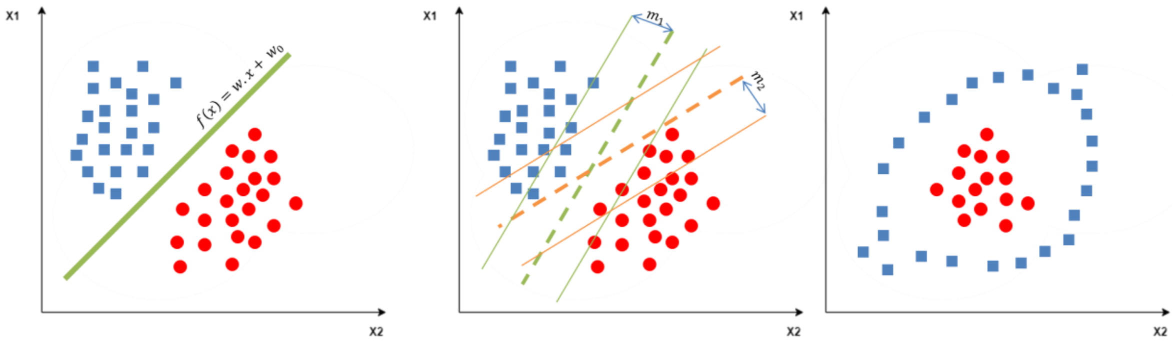


Figure Architecture of each kernel (linear, RBF, Polynomial) in Support Vector Machine

2.11 Conclusion :

In this chapter, we have overview some important notions, starting from digital image processing and deep learning, passing by features extraction and classification methods, the convolutional neural network, convolutional auto-encoder, finalizing by the support vector machine. Our goal was to provide some details on the work background, making it easy to understand the work and its context.

Chapter III:

Material and Methods

3.1 Introduction:

Our goal in this chapter is to present the steps of our work and our proposed method that comprises in four stages, the first step is using pre-train model Resnet-50 for extract feature from the dataset ,the second step is build our model CAE and use it from features extraction, the third step is using models of SVM for classification ,in end we use the fusion technique to improve it , as shown in the Fig 13.

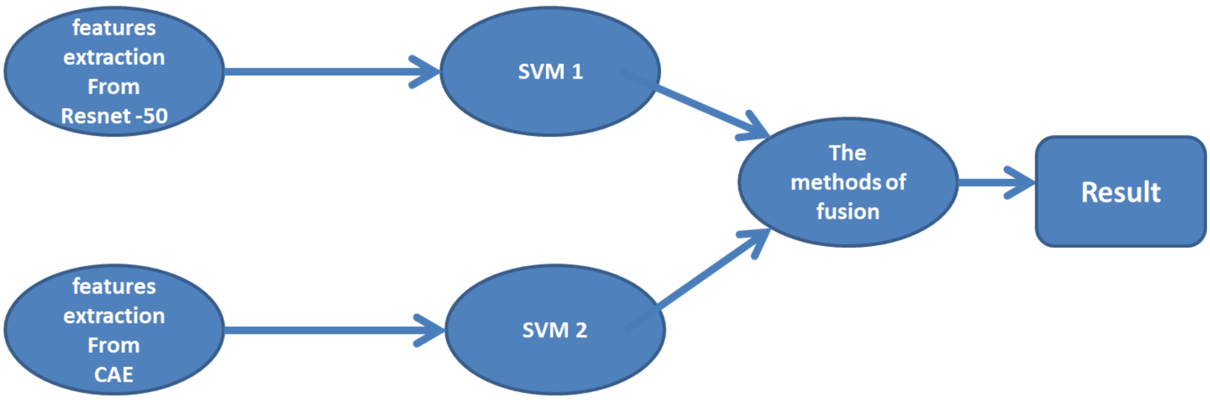


Figure : Our methods

3.2 Extracting Features Using RESNET50 :

ResNet:

ResNet stands for Residual Network, which is a network of neurons that has been built or more specifically has been innovative, by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun in 2015, the main different between resnet model and the traditional models is residual blocks[20]:

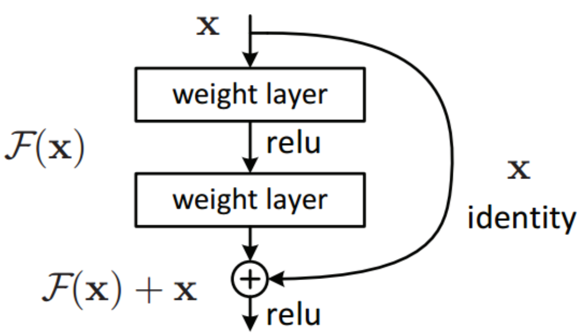


Figure : Residual block [20]

- how its works with the traditional models : each layer depend on the result of the activation function(relu for example) of the Z then[20]:

$$Z^{l+2} = W^{l+2} * X^{l+1} + B^{l+2}$$

- how its works with the resnet models : each layer depend on the result of the activation function(relu for example) of the Z plus the input of the layer before the prev [20]:

$$Z^{l+2} = W^{l+2} * X^{l+1} + B^{l+2\,+}\,X^{l}$$

Other Pre-Trained Models:

VGG19 :

- Built and trained by Karen Simonyan and Andrew Zisserman 2014
- Trained using more than 1 million images from the ImageNet database
- classify up to 1000 objects
- trained on 224x224 pixels colored images
- Number of Parameters: 143,667,240
- Depth: 26

Inceptionv3 :

- Build and trained by Google
- Trained by ImageNet database
- classify up to 1000 objects
- trained on 299x299 pixels colored images
- Number of Parameters: 23,851,784
- Depth: 159.

EfficientNet :

- Build and trained by Google 2019
- There are 8 alternative implementations of EfficientNet (B0 to B7)
- The simplest one, EfficientNetB0, is outstanding. With 5.3 million parameters
- Depth: 159

However, we chose Resnet-50 because it is much deeper than then other pre-trained models.

RESNET Architecture:

First, resnet started with a combination of [20]:

- The convolutional layers mostly have 3×3 filters and stride 2
- Total of 33 convolutional layers
- global average pooling layer and a 1000-class
- fully-connected layer with softmax
- The total number of weighted layers is 34

Second, resnet got a variant version each one of them has a specific architecture as:

- RESNET50
- ResNet-101
- ResNet-152

This section is about extracting features from the dataset images using resnet50 default architecture **Figure** [21]:

- Every ResNet architecture performs the initial convolution and max-pooling using 7×7 and 3×3 kernel sizes respectively
- ResNet50 has 4 stages as shown in the diagram **Figure**
- Stage 1 of the network starts and it has 3 Residual blocks containing 3 layers each
- Stage 2 of the network starts and it has 4 Residual blocks containing 3 layers each
- Stage 3 of the network starts and it has 6 Residual blocks containing 3 layers each
- Stage 4 of the network starts and it has 3 Residual blocks containing 3 layers each
- As we progress from one stage to another, the channel width is doubled and the size of the input is reduced to half

However, we excludes the softmax layer because our purpose is not to classify images in this step.

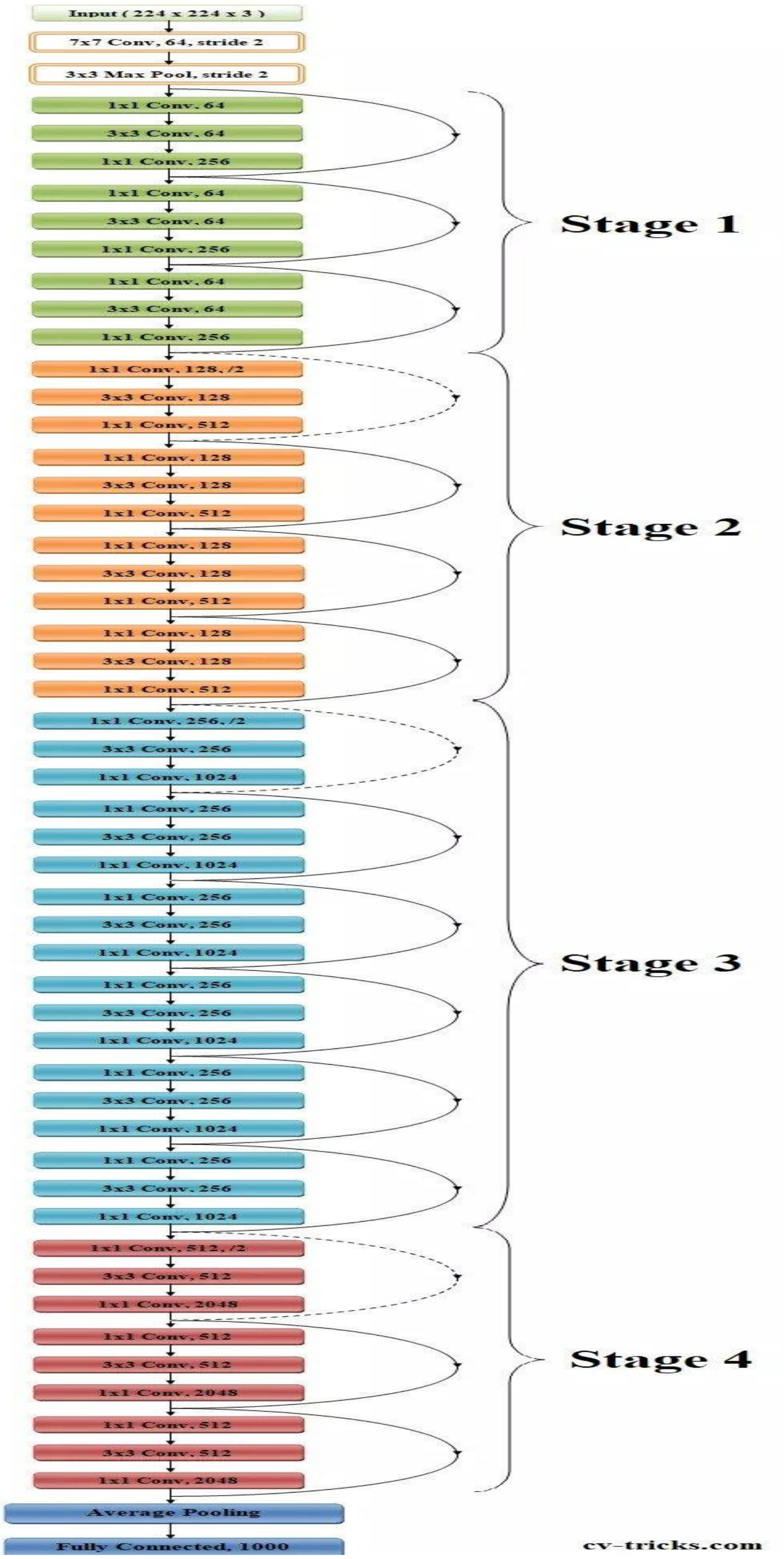


Figure the architecture of ResNet-50 [20

RESNET50 Implementation:

After taking the default architecture, we set several parameters:

- **Weights="imagenet":** that means our model will set its first weights values as the values after getting trained by imagenet dataset.
- include_top=false: as we said before this means our model will exclude the softmax layer.
- pooling="avg": average pooling is used to reduce the representation from say 7 x 7 x 2048 and average over all the spatial experiments to get 1x1x2048 [22].
- this means that we will not retrain the layers from resnet weights again

To clarify the model loop over the dataset by the "predict" method using weights from imagenet dataset with inputs was of shape 224×224×3, this process is executed using incremental learning (the batch size = 32) because it's helping us to avoid crush in the platform of google colab [23].

3.3 Convolutional Auto-Encoder (CAE) :

The goal of this section is how we used our model CAE for feature extraction and we based in the explication of the last chapter, our model CAE is used to reconstruct input images from feature extraction.

Our model is composed of two part encoder layer and decoder layer, in encoder layer is comprising by 6 convolution layer with (8,16,32,64,128,256) filters of kernel size 3x3, padding='same', stride=1, Relu function as activation function, in every convolution layer are follow by BatchNormalization layer and MaxPooling2D layer of shape 2x2, without padding and strides = 2.

In decoder layer is comprising by 6 convolution layer with (256,128,64,32,16,8) filters of kernel size 3x3, padding='same', stride=1, Relu as activation function, in every convolution 2D layer are follow by BatchNormalization layer and UpSampling2D layer of shape 2x2, in end we have convolution layer with 1 filters of kernel size 3x3, padding='same', stride=1, sigmoid function.

After training our model CAE, we use encoder layer to feature extraction of images with adding Flatten layer to be vector as size (4096x1), as shown in the tables below:

Layer (type)	Output Shape	Parameters
conv2d_1 (Conv2D)	(None, 256, 256,8)	80
batch_normalization (BatchNormalization)	(None, 256, 256,8)	32
max_pooling2d (MaxPooling2D)	(None, 128, 128,8)	0
conv2d_2 (Conv2D)	(None, 128,128,16)	1168
batch_normalization_1 (BatchNormalization)	(None, 128,128,16)	64
max_pooling2d_1 (MaxPooling2D)	(None, 64, 64, 16)	0
conv2d_3 (Conv2D)	(None, 64, 64, 32)	4640
batch_normalization_2(BatchNormalization)	(None, 64, 64, 32)	128
max_pooling2d_2(MaxPooling2D)	(None, 32, 32, 32)	0
conv2d_4 (Conv2D)	(None, 32, 32, 64)	18496
batch_normalization_3(BatchNormalization)	(None, 32, 32, 64)	256
max_pooling2d_3(MaxPooling2D)	(None, 16, 16, 64)	0
conv2d_5 (Conv2D)	(None, 16, 16,128)	73856
batch_normalization_4(BatchNormalization)	(None, 16, 16,128)	512
max_pooling2d_4(MaxPooling2D)	(None, 8, 8, 128)	0
conv2d_6 (Conv2D)	(None, 8, 8, 256)	295168
batch_normalization_5(BatchNormalization)	(None, 8, 8, 256)	1024
max_pooling2d_5(MaxPooling2D)	(None, 4, 4, 256)	0
Total parameters:	**395,424**	
Trainable parameters:	**394,416**	
Non-trainable parameters:	**1,008**	

Table Encoder layer (type layer and Output shape and number of parameters for each layer).

Layer (type)	Output Shape	Parameters
conv2d_7 (Conv2D)	(None, 4, 4, 256)	590080
batch_normalization_6 (BatchNormalization)	(None, 4, 4, 256)	1024
up_sampling2d (UpSampling2D)	(None, 8, 8, 256)	0
conv2d_8 (Conv2D)	(None, 8, 8, 128)	295040
batch_normalization_7 (BatchNormalization)	(None, 8, 8, 128)	512
up_sampling2d_1 (UpSampling2D)	(None, 16, 16,128)	0
conv2d_9 (Conv2D)	(None, 16, 16, 64)	73792
batch_normalization_8(BatchNormalization)	(None, 16, 16, 64)	256
up_sampling2d_2 (UpSampling2D)	(None, 32, 32, 32)	0
conv2d_10 (Conv2D)	(None, 32, 32, 32)	18464
batch_normalization _9 (BatchNormalization)	(None, 32, 32, 32)	128
up_sampling2d_3 (UpSampling2D)	(None, 64, 64, 32)	0
conv2d_5 (Conv2D)	(None, 64, 64, 16)	4624
batch_normalization_10 (BatchNormalization)	(None, 64, 64, 16)	64
up_sampling2d_4 (UpSampling2D)	(None, 128,128,16)	0
conv2d_6 (Conv2D)	(None, 128, 128,8)	1160
batch_normalization _11 (BatchNormalization)	(None, 128, 128,8)	32
up_sampling2d_5 (UpSampling2D)	(None, 256, 256,8)	0
conv2d_13 (Conv2D)	(None,256,256,1)	73
Total parameters:	**985,249**	
Trainable parameters:	**984,241**	
Non-trainable parameters:	**1,008**	

Table Decoder layer (type layer and Output shape and number of parameters for each layer).

Layer (type)	Output Shape	Parameters
conv2d_input (InputLayer)	(None, 256, 256,1)	0
conv2d (Conv2D)	(None, 256, 256,8)	80
max_pooling2d (MaxPooling2D)	(None, 128, 128,8)	0
conv2d_1 (Conv2D)	(None, 128,128,16)	1168
max_pooling2d_1 (MaxPooling2D)	(None, 64, 64, 16)	0
conv2d_2 (Conv2D)	(None, 64, 64, 32)	4640
max_pooling2d_2(MaxPooling2D)	(None, 32, 32, 32)	0
conv2d_3_4 (Conv2D)	(None, 32, 32, 64)	18496
max_pooling2d_3(MaxPooling2D)	(None, 16, 16, 64)	0
conv2d_5 (Conv2D)	(None, 16, 16,128)	73856
max_pooling2d_4(MaxPooling2D)	(None, 8, 8, 128)	0
conv2d_6 (Conv2D)	(None, 8, 8, 256)	295168
max_pooling2d_5(MaxPooling2D)	(None, 4, 4, 256)	0
sequential_1 (Sequential)	(None, 256, 256,1)	983233
Total parameters:	**1,376,641**	
Trainable parameters:	**1,376,641**	
Non-trainable parameters:	**0**	

Table Model CAE (type layer and Output shape and number of parameters for each layer).

Layer (type)	Output Shape	Parameters
conv2d_input (InputLayer)	(None, 256, 256,1)	0
conv2d (Conv2D)	(None, 256, 256,8)	80
max_pooling2d (MaxPooling2D)	(None, 128, 128,8)	0
conv2d_1 (Conv2D)	(None, 128,128,16)	1168
max_pooling2d_1 (MaxPooling2D)	(None, 64, 64, 16)	0
conv2d_2 (Conv2D)	(None, 64, 64, 32)	4640
max_pooling2d_2(MaxPooling2D)	(None, 32, 32, 32)	0
conv2d_3_4 (Conv2D)	(None, 32, 32, 64)	18496
max_pooling2d_3(MaxPooling2D)	(None, 16, 16, 64)	0
conv2d_5 (Conv2D)	(None, 16, 16,128)	73856
max_pooling2d_4(MaxPooling2D)	(None, 8, 8, 128)	0
conv2d_6 (Conv2D)	(None, 8, 8, 256)	295168
max_pooling2d_5(MaxPooling2D)	(None, 4, 4, 256)	0
flatten (Flatten)	(None, 4096)	0
Total parameters:	**393,408**	
Trainable parameters:	**393,408**	
Non-trainable parameters:	**0**	

Table Model feature extraction Encoder layer with Flatten layer (type layer and Output shape and number of parameters for each layer).

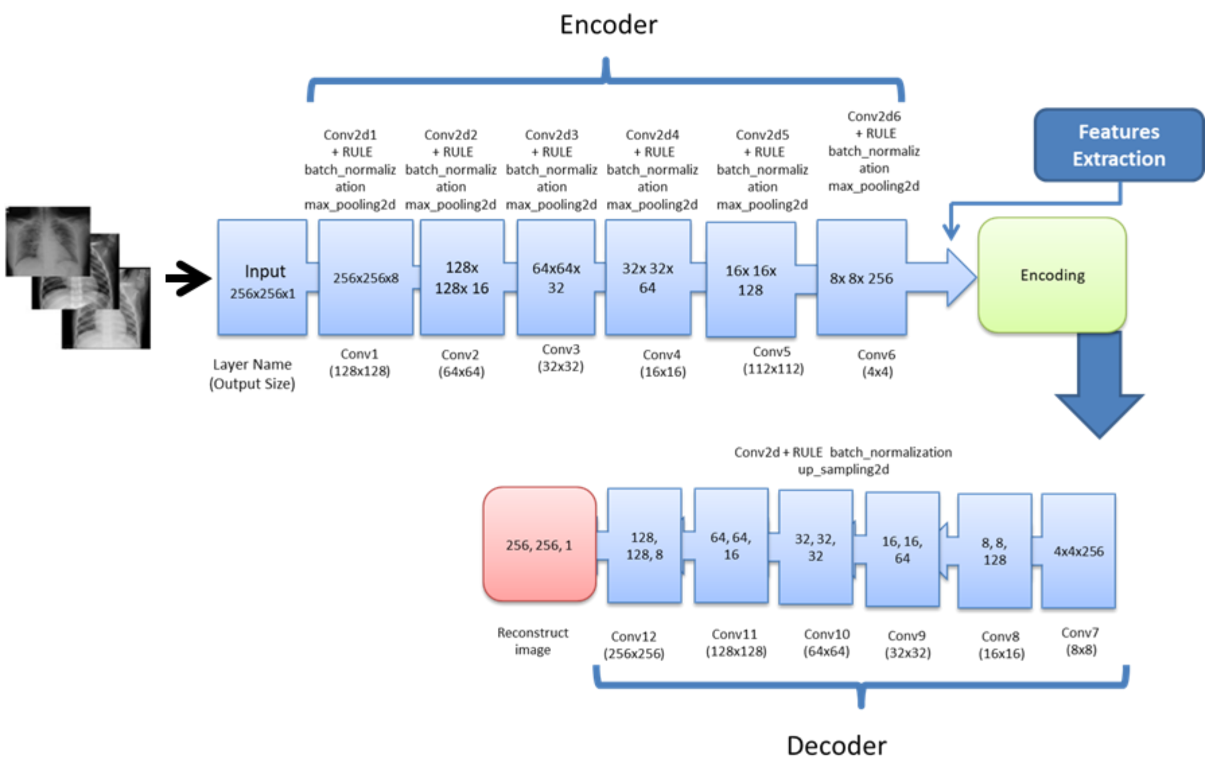


Figure : Architecture of the proposed CAE

Our model CAE, it can be summarized as follows:

- Train our model CAE with a dataset.
- Build model of feature extraction by using pre train encoder layer of our model and adding Flatten layers.
- Using model of feature extraction to extract all features from images in dataset.

In the end, we aim to advantages of using Convolutional Auto-Encoder:

- Convolutional Auto-Encoder model are compatible with ResNet-5 (CNN), we proof this in chapter of result.
- Convolutional Auto-Encoder model is one of unsupervised learning models; it solves the problem by learning without any labels.
- Convolutional Auto-Encoder model is easy and fast in training because unsupervised learning is less complexity in comparison with supervised learning.

3.4 Support Vector Machine (SVM) :

The goal of this section is how we used our model SVM for classification images from dataset and we based in the explication of the last chapter, our model SVM is used to classifier by the feature extraction of images .

In our model, we use two models of SVM with same kernel (Linear and Radial basis function also known as RBF) and same function other for one (ovo) , in kernel RBF we used same hyper parameters C (c=10).

After prepare our models, we train it by using feature of images from our models ResNet-50 and CAE.

Our model SVM, it can be summarized as shown in the Fig 17:

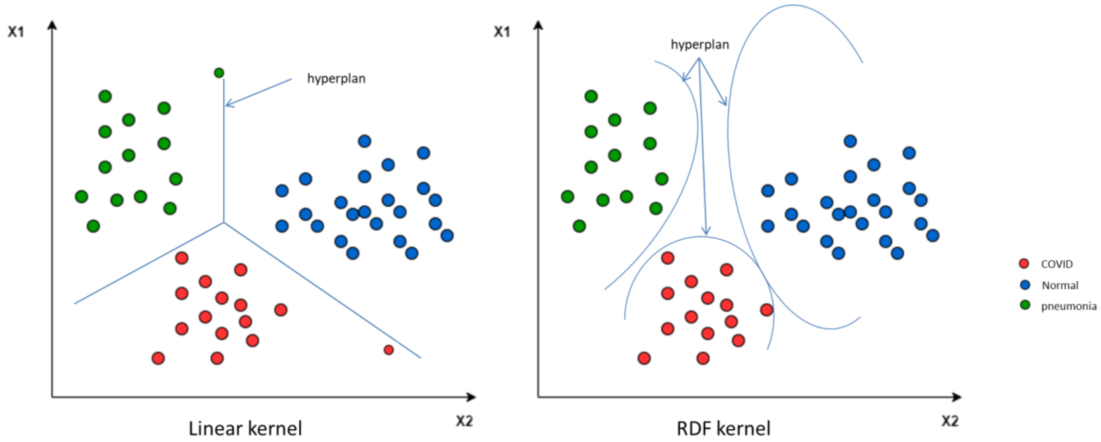


Figure : our SVM model

In the end, we aim to advantages of using Support Vector Machine (SVM):

- The kernel trick is real strength of SVM. With an appropriate kernel function, we can solve some complex problem.
- Unlike neural networks, SVM is not solved for local optima.
- It adapts relatively well to high dimensional data.
- SVM models have a generalisation in practice, the risk of over-adjustment is reduced in SVM.
- SVM is always compared with ANN. When compared to ANN models, SVMs give better results.

3.5 The fusion technique:

In this section, we aim to how we used the methods of fusion in our models and it used to improve our models.

- We take the output of our models as probability and labels, our models are four models and we based in the explication of the last part:
- The first one models is SVM with linear kernel and input data as feature of images from our model CAE and we refer to as **S1**.
- The second models are SVM with linear kernel and input data as feature of images from our model ResNet-50 and we refer to as **S2**.
- The third models are SVM with RBF kernel and input data as feature of images from our models CAE, we refer to as **S3**.
- The last fourth model is SVM with RBF kernel and input data as feature of images from our model ResNet-50, we refer to as **S4**.

After this we apply the three methods of fusion, as follows:

Score level fusion (max, min, sum, product ,mean):

In this method, we used in two way the first one is take probability from S1 and S2 and in the second we used probability from S1 and S2, S3, S4.

The output probability is probability of belonging each class, the output is represented as array of size $3032 * 3$, in each 3 column represents the class symbolized by *m* and 3032 rows is number of test images symbolized by *n*.

The output probability of each model is Symbolized by SPi and size $(n * m)$

Over the last explication, now we take look about formula of each function of score level fusion $(max, min, sum, product\ , mean)$ symbolized by :

$$(fmax, fmin, fsum, fprudoct, fmean)$$

In score level fusion with S1, S2 for each n:

$$\boldsymbol{fmax} = \sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} [\mathbf{max}(\sum_{\boldsymbol{i=1}}^{\mathbf{2}} \boldsymbol{SP_i}[\boldsymbol{j}, \mathbf{1}]) , \mathbf{max}(\sum_{\boldsymbol{i=1}}^{\mathbf{2}} \boldsymbol{SP_i}[\boldsymbol{j}, \mathbf{2}]) , \mathbf{max}(\sum_{\boldsymbol{i=1}}^{\mathbf{2}} \boldsymbol{SP_i}[\boldsymbol{j}, \mathbf{3}])]$$

$$\boldsymbol{fmin} = \sum_{j=1}^{n}[\boldsymbol{min}(\sum_{i=1}^{2}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{1}])\,,\boldsymbol{min}(\sum_{i=1}^{2}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{2}])\,,\boldsymbol{min}(\sum_{i=1}^{2}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{3}])]$$

$$\boldsymbol{fsum} = \sum_{j=1}^{n}[\sum_{i=1}^{2}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{1}]\,,\sum_{i=1}^{2}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{2}]\,,\sum_{i=1}^{2}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{3}]]$$

$$\boldsymbol{fproduct} = \sum_{j=1}^{n}[\prod_{i=1}^{2}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{1}]\,,\prod_{i=1}^{2}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{2}]\,,\prod_{i=1}^{2}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{3}]]$$

$$\boldsymbol{fmean} = \sum_{j=1}^{n}[(\prod_{i=1}^{2}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{1}]) \div \boldsymbol{2}\,,(\prod_{i=1}^{2}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{2}]) \div \boldsymbol{2}\,,(\prod_{i=1}^{2}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{3}]) \div \boldsymbol{2}]$$

In score level fusion with S1, S2, S3, S4 for each n:

$$\boldsymbol{fmax} = \sum_{j=1}^{n}[\mathbf{max}(\sum_{i=1}^{4}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{1}])\,,\mathbf{max}(\sum_{i=1}^{4}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{2}])\,,\mathbf{max}(\sum_{i=1}^{4}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{3}])]$$

$$\boldsymbol{fmin} = \sum_{j=1}^{n}[\boldsymbol{min}(\sum_{i=1}^{4}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{1}])\,,\boldsymbol{min}(\sum_{i=1}^{4}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{2}])\,,\boldsymbol{min}(\sum_{i=1}^{4}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{3}])]$$

$$\boldsymbol{fsum} = \sum_{j=1}^{n}[\sum_{i=1}^{4}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{1}]\,,\sum_{i=1}^{4}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{2}]\,,\sum_{i=1}^{4}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{3}]]$$

$$\boldsymbol{fproduct} = \sum_{j=1}^{n}[\prod_{i=1}^{4}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{1}]\,,\prod_{i=1}^{4}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{2}]\,,\prod_{i=1}^{4}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{3}]]$$

$$\boldsymbol{fmean} = \sum_{j=1}^{n}[(\prod_{i=1}^{4}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{1}]) \div \boldsymbol{4}\,,(\prod_{i=1}^{4}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{2}]) \div \boldsymbol{4}\,,(\prod_{i=1}^{4}\boldsymbol{SP_i}[\boldsymbol{j},\boldsymbol{3}]) \div \boldsymbol{4}]$$

In the end, we apply function to take label by take index of Max probability in 3 columns.

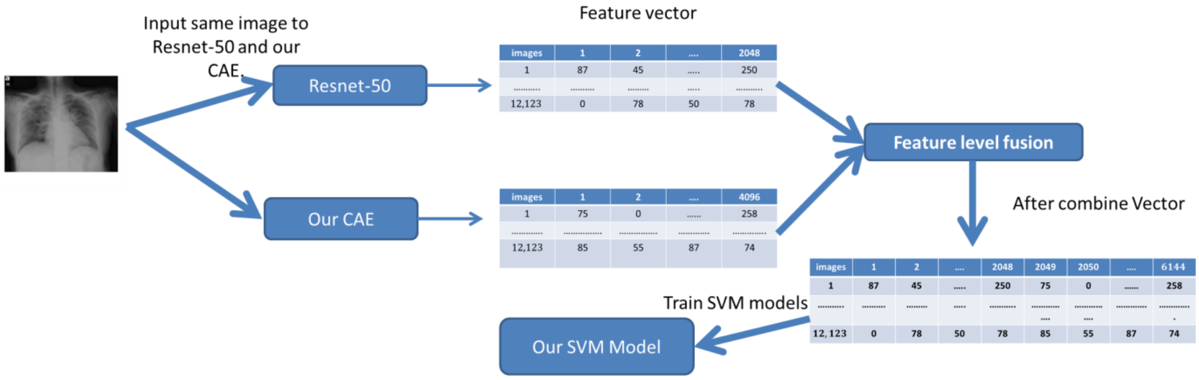
Feature level fusion:

In this method, we combine features of images from our models ResNet-50 and CAE after pre-process dataset and we train it by two models SVM linear kernel and RBF kernel .

In the features of images dataset from our models ResNet-50 size is$[12{,}123 * 2048]$.

In the features of images dataset from our models CAE size is$[12{,}123 * 4096]$.

In the end, The result of combining size is$[12{,}123 * 6144]$, it can be summarized as shown in the Fig 18:



images	1	2	....	2048
1	87	45	.....	250
....	....	....	.....	....
12,123	0	78	50	78

images	1	2	....	4096
1	75	0	......	258
....	....	....	....	....
12,123	85	55	87	74

images	1	2	....	2048	2049	2050	....	6144
1	87	45	.....	250	75	0	......	258
....	....	....	.....	....	....	....	....	....
12, 123	0	78	50	78	85	55	87	74

Figure : Feature level fusion

Decision level fusion:

In this method, also known as voting function or majority voting, we use in this technique output label from SVM models (S1, S2, S3, S4).

Voting function work by choice most big number of label about image but if the number of labels is similar we choice Random label.

3.6 Conclusion :

In this chapter, we have look about our methods that we used in our work, the proposed method is presented, where its three stages, namely preparation dataset, feature extraction using our models Resnet-50 and CAE, in the classification we used SVM models, for improve our work we use the fusion technique.

Chapter IV:

Experimental Results

This chapter is to report our findings. We perform experiments that measure the proposed method from different aspects. We start by presenting our framework and process of collection dataset and the performance metrics. Then, experimental results are reported.

4.1Our Framework:

In our framework, we use python version 3.9 as Programming Language and we use site GOOGLE celebratory (Colab)as a great platform because it offers a free powerful resource for train and deep learning runs, in our work we use Colab with 12GB NVIDIA Tesla K80 GPU and 18GB RAM and CPU Intel (R) Xeon (R) 2.30GHz.

4.2 Collection of Dataset:

For the training and evaluation of our model we use dataset from COVID-19 radiography database (Located in the kaggel platform) [24], this dataset contains a radiographic image (CXR) for four categories (Normal, Covid, Lung Opacity, Viral Pneumonia) :

The class	Number of images
COVID-19	3616
Viral Pneumonia	1345
Normal	10200
Lung Opacity	6012
Total	21173

Table COVID-19 Radiography Dataset statistics

Then we choose only three classes (Normal, Covid-19, Viral Pneumonia) due to compare it with Result of other Work, in addition we divided the content of each class of dataset for the training group and the test group:

The class	Number of Test images	Number of train images	Number of images
COVID-19	724	2169	3616
viral pneumonia	269	807	1345
normal	2039	6115	8861
Total	**3032**	**9091**	**12123**

Table our dataset statistics

4.3 Experimental protocol:

Performance metric:

To calculate the performance of the proposed model, Recall and Precision, F1-Score ,Accuracy metrics are adopted, which are depicted in the form of Confusion matrix (CM), which are given as follows**:**

$$\mathbf{Recall} = \frac{\boldsymbol{Sum\ of\ all\ True\ Positives}}{\boldsymbol{Sum\ of\ all\ True\ Positives + Sum\ of\ all\ False\ Negatives}}$$

$$\mathbf{Precision} = \frac{\boldsymbol{Sum\ of\ all\ True\ Positives}}{\boldsymbol{Sum\ of\ all\ True\ Positives + Sum\ of\ all\ False\ Positives}}$$

$$\mathbf{F1 - Score} = \frac{\mathbf{2 \times Recall \times Precision}}{\mathbf{Recall + Precision}}$$

$$\mathbf{Accuracy} = \frac{\boldsymbol{Number\ of\ test\ images\ correct\ classified}}{\boldsymbol{Total\ number\ of\ test\ images}}$$

4.4 The proposed method Results:

4.4.1 RESNET50 Results:

Because of our model loop over each image from the dataset will give us a vector [1 * 2049], the first column is for the labels [0, 1, or 2] and the rest is the features that were extracted from this image.

In conclusion, that will give us a vector of [9091 * 2049] for the training images and a vector of [3032 * 2049] for the test images each one stored in a CSV file temporarily.

4.4.2. First Result: Convolutional Auto-Encoder (CAE):

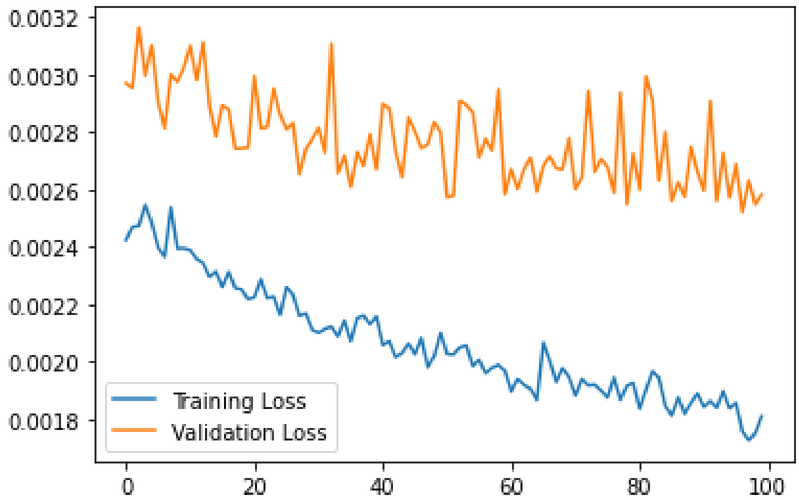


Figure the Training loss function and Validation loss reached by each epoch.

From the above Figure , we can clearly see that line of training loss function converges rapidly to 0.0015 but after epoch number 95 is back rapidly 0.0018 and in validation loss after each epoch , value not stabilizing but after epoch number 85 value stabilizing in 0.0026.

Models	Estimation (Minute)
Feature extraction using Resnet-50	127 Minute
CAE train	280 Minute
Feature extraction using CAE	60 Minute
SVM (S1)	4,5Minute
SVM (S2)	4 Minute
SVM (S3)	7.37 Minute
SVM (S4)	5 Minute
Feature level fusion with model SVM with kernel rbf	7 Minute
SVM Feature level fusion with model SVM with kernel linear	10 Minute
Decision level fusion	12 Minute

Table Estimation for each Model

From the above table, we found the process of feature extraction from dataset take long time but classification is short time.

4.4.3. Second Result: SVM (S1, S2, S3, S4)

Here the table, we can see Confusion matrix (CM) SVM linear with CAE (S1) and accuraccy of this model is 95%.

Class	Precision	recall	F1-score	support
COVID-19	**0.90**	**0.91**	**0.90**	**724**
Normal	**0.96**	**0.96**	**0.96**	**2039**
Viral Pneumonia	**0.94**	**0.94**	**0.94**	**269**
Accuracy			**0.95**	**3032**

Table the Confusion matrix (CM) SVM linear with CAE (S1)

Here the table, we can see Confusion matrix (CM) SVM linear with Resnet-50 (S2) and accuraccy of this model is 96% and it's better than the previous result.

Class	Precision	recall	F1-score	support
COVID-19	**0.93**	**0.93**	**0.93**	**724**
Normal	**0.97**	**0.97**	**0.97**	**2039**
Viral Pneumonia	**0.96**	**0.96**	**0.96**	**269**
Accuracy			**0.96**	**3032**

Table the Confusion matrix (CM) SVM linear with Resnet-50 (S2)

Here the table, we can see Confusion matrix (CM) SVM RBF kernel with CAE (S3) and accuraccy of this model is 97% and it's better than the two previous result.

Class	Precision	Recall	F1-score	support
COVID-19	**0.95**	**0.93**	**0.94**	**724**
Normal	**0.97**	**0.98**	**0.98**	**2039**
Viral Pneumonia	**0.96**	**0.96**	**0.96**	**269**
Accuracy			**0.97**	**3032**

Table the Confusion matrix (CM) SVM RBF kernel with CAE (S3)

Here the table, we can see Confusion matrix (CM) SVM RBF kernel with Resnet-50 (S4) and accuraccy of this model is 96% and it's bad than the previous result and it's same than the frist previous result.

Class	Precision	Recall	F1-score	support
COVID-19	**0.97**	**0.89**	**0.93**	**724**
Normal	**0.96**	**0.99**	**0.97**	**2039**
Viral Pneumonia	**0.97**	**0.94**	**0.96**	**269**
Accuracy			**0.96**	**3032**

Table the Confusion matrix (CM) SVM RBF kernel with Resnet-50 (S4)

4.4.4. Third Result: The technique of Feature level fusion with model SVM with kernel RBF and linear:

Here the table, we can see Confusion matrix (CM) Feature level fusion with SVM RBF kernel and accuraccy of this model is 98% and it's better than all the previous result.

Class	Precision	Recall	F1-score	support
COVID-19	**0.97**	**0.97**	**0.97**	**724**
Normal	**0.99**	**0.98**	**0.99**	**2039**
Viral Pneumonia	**0.95**	**0.98**	**0.97**	**269**
Accuracy			**0.98**	**3032**

Table the Confusion matrix (CM) Feature level fusion with SVM RBF kernel

Here the table, we can see Confusion matrix (CM) Feature level fusion with SVM linear kernel and accuraccy of this model is 97% and it's bad than the previous result.

Class	Precision	Recall	F1-score	support
COVID-19	**0.97**	**0.96**	**0.96**	**724**
Normal	**0.98**	**0.99**	**0.98**	**2039**
Viral Pneumonia	**0.94**	**0.97**	**0.96**	**269**
Accuracy			**0.97**	**3032**

Table the Confusion matrix (CM) Feature level fusion with SVM linear kernel

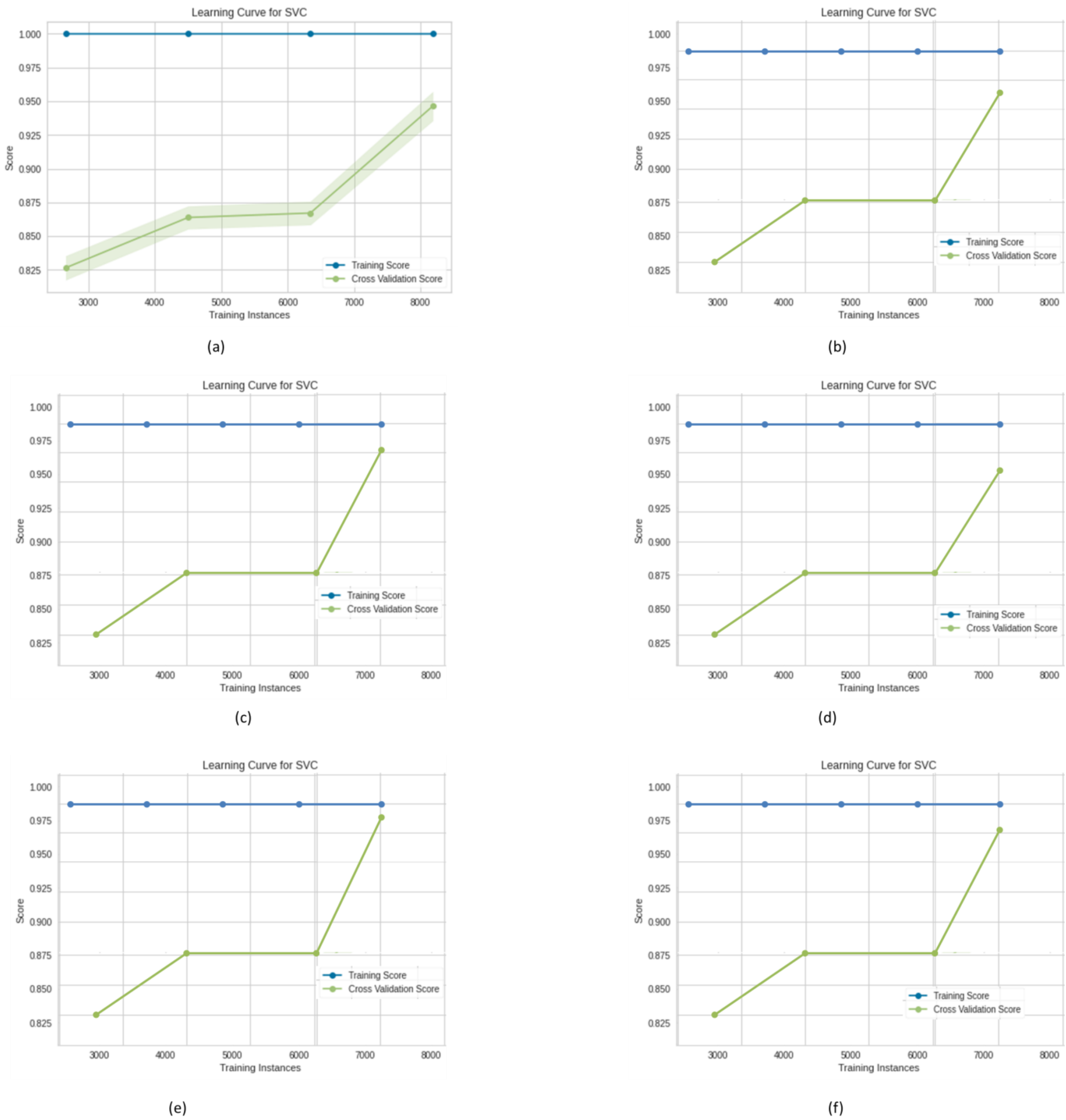


Figure the learning curve of our models SVM by the training score and the cross-validation score

From the above Figure, we see our models (Fig.a is S1, Fig.b is S2, Fig.c is S3, Fig.d is S4 , Fig.e and Fig.f is Feature level fusion and SVM RBF and linear) the training score and the cross-validation score are both very good at the end. We can see clearly that the training score is still around the maximum and the validation score could be increased with more training samples.

4.4.5. Total Result: The fusion technique:

Here the table, we can see total result the fusion technique and accuraccy of each method, best accuraccy method is score level fusion (sum, product, mean) with four model SVM (s1,s2,s3,s4) by 98.74% and it's better than all the result.

The methods	Result (accuracy)
Model SVM kernel linear with CAE(s1)	95%
Model SVM kernel linear with CNN(s2)	96%
Score level fusion (max,min,sum,prodoct,mean) with two model SVM (s1,s2):	Max:97.92% Min: 97.92% Sum: 97.92% Product: 97.92% Mean: 97.92%
New model of SVM with kernel RBF with CAE(s3):	97%
New model of SVM with kernel RBF with CNN(s4):	96%
Score level fusion (max,min,sum,product,mean) with four model SVM (s1,s2,s3,s4):	Max:98.31% Min: 98.41% Sum: 98.74% Product: 98.74% Mean: 98.74%
The technique of Feature level fusion with model SVM with kernel rbf:	98%
The technique of Feature level fusion with model SVM kernel linear:	97%
Decision level fusion **Voting function (majority voting) with four svm model (s1,s2,s3,s4)**	97%

Table total result the fusion technique

4.4.6 Performance comparison with other works:

Here the table, we can see comparison accuracy of performance measures with other works, best accuracy method is our method score level fusion (sum,product,mean) with four model SVM linear kernel(s1,s2,s3,s4) by 98.74% and it's better than some the work .

The methods	Result (accuracy)
The technique of Feature level fusion with model SVM with kernel rbf	**98%**
Score level fusion (sum,prodoct,mean) with four model SVM linear kernel(s1,s2,s3,s4)	**98.74%** **98.74%** **98.74%**
COVIDetection-Net[6]	**99.72 %**
COV-19Ne[7]	**99.45%**
COVID-19 Detection [8]	**99.9%**
ResNet50-SVM [9]	**98.66 %**

Table comparison accuracy of performance measures with other works

4.4.7 System form:

Here we are going to present interfaces of our system in site web.

Upload page:

In this figure Fig. , we present upload image to process it:

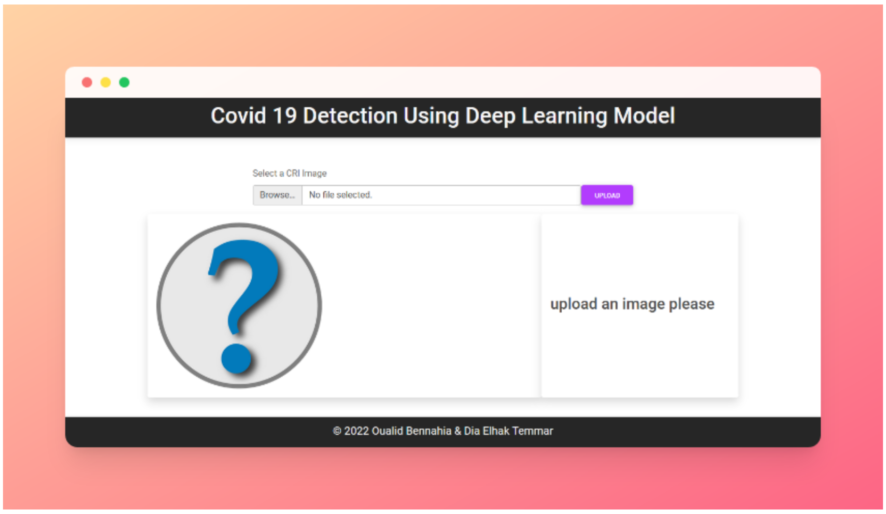


Figure : Upload page

Result page:

In this figure Fig. and Fig. , Fig. , we present result of image by using our best model fusion (score level fusion (sum, product, mean))four SVM in each case:

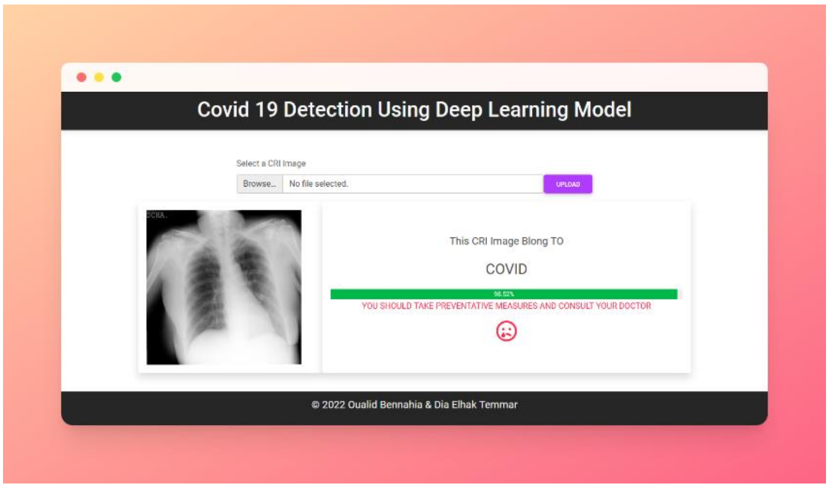


Figure : result of model is COVID

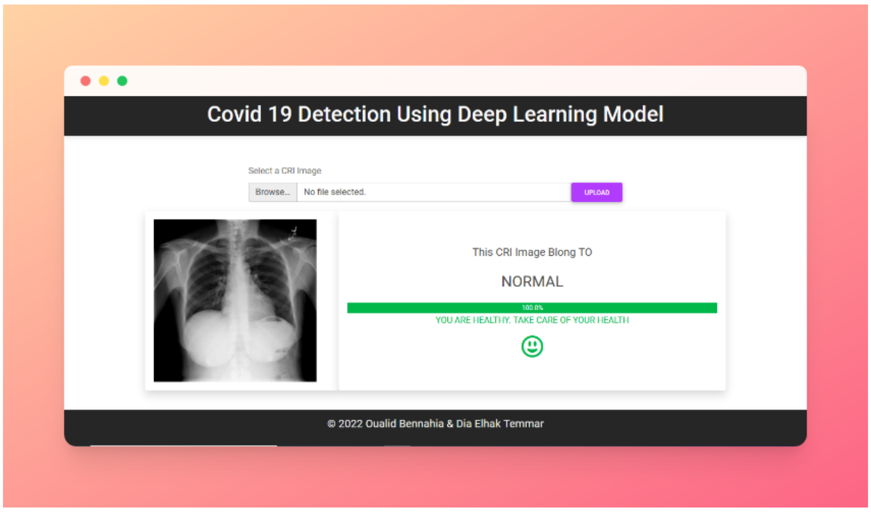


Figure : result of model is Normal

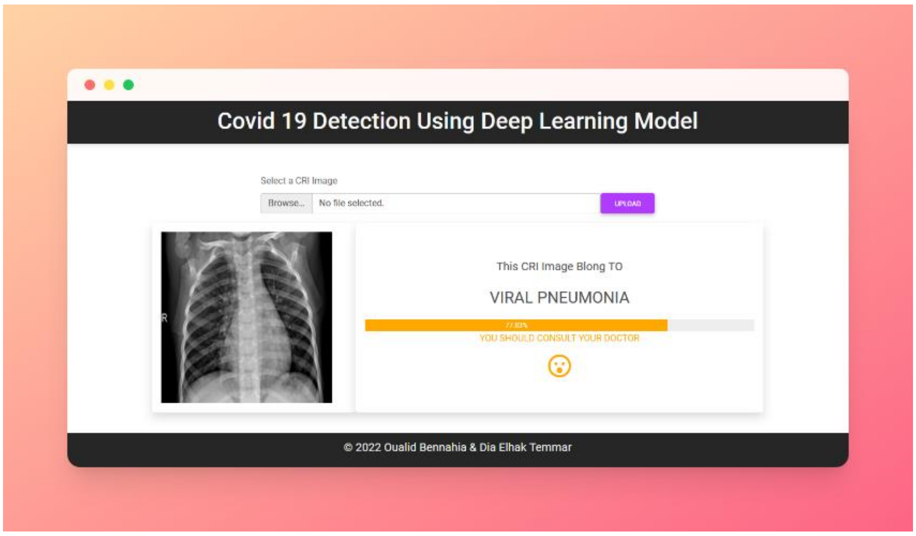


Figure : Result of model is viral pneumonia

4.5 Conclusion:

In this chapter, we have reported our framework and preparation of our collection of the dataset, the experimental results of our methods that we are carrying out. We have also studied the effect of The fusion technique. At the end, our methods are compared with other work in deep learning and we show our system form of site web and how is it work.

General Conclusion

Deep learning is a very big area but it has big popularity in the last five years by researcher and big enterprises because deep learning is Important part in Machine learning and also in artificial intelligent systems .Also, we found deep learning in many fields, such as Remote sensing , Healthcare, Intelligent transportation....etc.

We can find several operations of the human brain that were simulated by computers, such as knowledge, learning, reasoning, recognition and classification…etc.

In this thesis, we have designed a system for detection Covid-19 from chest radiographic images using deep learning techniques. Our aim was to build a classifier based on our model.

Our method can be divided into four phases, prepare our dataset, features extraction from dataset using our models Resnet-50 and CAE, classification based on our model SVM and improve it by using fusion technique.

Experiments result was carried out with our models, with accuracy as a performance metric .Absolutely, our findings were good Result for us.

The experimental results show that the proposed model had the best performance, accuracy but Compared to other Models, has close to other Models.

However, our model is a commercial-ready solution, but our hopes, in First place is that the promising results obtained from this work can be leveraged and further be improvised by both Master students and researchers to accelerate the development of highly accurate deep learning solutions for detecting COVID-19.

In addition, many other detecting is possible for future works, like develop model can be generated heat map to find the location of illness or injury in images.

Bibliography

[1] Coronavirus disease (COVID-19) pandemic. (n.d.). Retrieved april 18, 2022, from world health organization: https://www.euro.who.int/en/health-topics/health-emergencies/coronavirus-covid-19/novel-coronavirus-2019-ncov

[2] WHO Coronavirus (COVID-19) Dashboard. (2022, June 10). Retrieved from world health organization: https://covid19.who.int/

[3] COVID-19 Testing. (2022, February 25). Retrieved from centers for disease control and preventio: https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/testing.html

[4] Andŕes, A.-I., Leonel , M.-J., & Martha , Z.-D. (2021). An overview of deep learning in medical imaging. Informatics in Medicine Unlocked, 26. Retrieved from https://www.sciencedirect.com/science/article/pii/S2352914821002033?via%3Dihub

[5] Education, I. C. (2020, May 1). Deep Learning. Retrieved from IBM: https://www.ibm.com/cloud/learn/deep-learning

[6] T.K.S., R.M.D., & H.K.K. (2019). Lung Cancer Detection Based on CT Scan Images by Using Deep Transfer Learnin. IIETA, 36(4), 339–344. https://www.iieta.org/journals/ts/paper/10.18280/ts.360406

[7] C.M.-I., V.M.C., C.I., Z.R.-E., B.B., S.E.P., & K.L. (2020, January 24). Image-Based Cardiac Diagnosis With Machine Learning: A Review. Frontiersin. Retrieved June 7, 2022, from https://www.frontiersin.org/articles/10.3389/fcvm.2020.00001/full

[8] Afza, F.; Sharif, M.; Khan, M.A.; Tariq, U.; Yong, H.-S.; Cha, J. Multiclass Skin Lesion Classification Using Hybrid Deep Features Selection and Extreme Learning Machine. Sensors 2022, 22, 799.https://doi.org/10.3390/s22030799

[9] Dai, L., Wu, L., Li, H., Cai, C., Wu, Q., Kong, H., Liu, R., Wang, X., Hou, X., Liu, Y., Long, X., Wen, Y., Lu, L., Shen, Y., Chen, Y., Shen, D., Yang, X., Zou, H., Sheng, B., & Jia, W. (2021, May). A deep learning system for detecting diabetic retinopathy across the disease spectrum (No. 3242). nature communications. https://www.nature.com/articles/s41467-021-23458-5

[10] S.Elkorany, A., & F.Elsharkawyc, Z. (2021). COVIDetection-Net: A tailored COVID-19 detection from chest radiography images using deep learning. Optik, 231. https://doi.org/10.1016/J.IJLEO.2021.166405

[11] Asif, S., & Amjad, K. (2020). Automatic COVID-19 Detection from chest radiographic images using Convolutional Neural Network JF - medRxiv. medRxiv. https://doi.org/10.1101/2020.11.08.20228080

[12] D. Das, K.C. Santosh, U. Pal, Truncated inception net: COVID-19 outbreak screening using chest X-rays, Phys. Eng. Sci. Med. (2020) no. 0123456789

[13] P.K. Sethy, S.K. Behera, P.K. Ratha, P. Biswas, Detection of coronavirus disease (COVID-19) based on deep features and support vector machine, Int. J. Math. Eng. Manage. Sci. 5 (4) (2020) 643–651.

[14] Rashid, N., Faisal Hossain, M. A., Ali, M., Sukanya, M. I., Mahmud, T., & Fattah, S. A. (2021). AutoCovNet: Unsupervised feature learning using autoencoder and feature merging for detection of COVID-19 from chest X-ray images. Biocybernetics and Biomedical Engineering, 41(4), 1685–1701. https://doi.org/10.1016/J.BBE.2021.09.004

[15] Gonzales, R. a. (2002, Prentice hal). Digital image processing.l.

[16] Ian Goodfellow, Y. B. (Nov 18, 2016 .). Deep Learning , . MIT Press,.

[17] Miika Silfverberg. ((Chap. 6), 2007). in Text Entry Systems, .

[18] Cortes, C., & Vapnik, V. N. ((1995)). "Support-vector networks“.

[19] Data files Original Authors. (2021). COVID-19 Radiography Database. Kaggle. Retrieved March 10, 2022, from https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database

[20] He, K., Zhang, X., Ren, S., & Sun, J. (2016; , June 27–30). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,. Las Vegas, NV, USA, , pp. 770–778.

[21] Kaiming He, X. Z. (n.d.). Identity Mappings in Deep Residual Networks.

[22] Explain Pooling layers:. (2021, June 26). Retrieved from Max Pooling, Average Pooling, Global Average Pooling, and Global Max pooling. Retrieved from knowledge

Transfer: https://androidkt.com/explain-pooling-layers-max-pooling-average-pooling-global-average-poo

[23] Rosebrock, A. (2019, May 27). Keras: Feature extraction on large datasets with Deep Learning. : Retrieved from pyimagesearch: https://pyimagesearch.com/2019/05/27/keras-feature-extraction-on-large-datasets-with-deep-learning/

[24] Data files Original Authors.(2021). (. Retrieved March 10, 2022,). COVID-19 Radiography Database. Retrieved from Kaggle: https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database