People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Larbi Tébessi University - Tébessa
Faculty of Exact Sciences and Natural and Life Sciences
Department: Mathematics and Computer Science

Final thesis
For obtaining of the *MASTER* diploma
Domain: Mathematics and Computer Science
Field: Computer Science
Specialty: Systems and Multimedia

IMAGE CLASSIFICATION USING TEXTURAL FEATURES

Topic submitted by:

Guerdi Nabil

Before the jury:

Dr. Bennour Akram	MCA	Univérsité Larbi Tébessi	President
Dr. Bendib Issam	MCB	University Larbi Tébessi	Examiner
Dr. Gattal Abdeljalil	MCA	University Larbi Tébessi	Thesis Supervisor

Date of defense: 21 June / 2021

Abstract

Image classification is a classic problem in the fields of image processing, computer vision and machine learning. The purpose of the classification process is to classify all pixels in a digital image into one of several categories. The goal of image classification is to identify the features of certain object using different images for the same object. This helps to identify similarities between the same objects. Therefore, we will be able to use these features to classify images.

In this work we will try to make an image classification system using three different datasets (CIFAR-10, CIFAR-100, STL-10) with two different approaches with highest accuracy rate possible.

Keywords: Image classification, image processing, computer vision, machine learning, features computing.

Résumé

La classification d'images est un problème classique dans les domaines du traitement d'images, de la vision par ordinateur et de l'apprentissage automatique. Le but du processus de classification est de classer tous les pixels d'une image numérique dans l'une de plusieurs catégories. Le but de la classification d'images est d'identifier les caractéristiques de certains objets en utilisant différentes images pour le même objet. Cela permet d'identifier les similitudes entre les mêmes objets. Par conséquent, nous pourrons utiliser ces fonctionnalités pour classer les images.

Dans ce travail, nous essaierons de créer un système de classification d'images en utilisant trois ensembles de données différents (CIFAR-10, CIFAR-100, STL-10) avec deux approches différentes avec le taux de précision le plus élevé possible.

Mots clés : La classification des images, la vision par ordinateur, l'apprentissage automatique, les pixels, image numérique.

الملخص

يعد تصنيف الصور مشكلة كلاسيكية في مجالات معالجة الصور والرؤية الحاسوبية أيضا التعلم الآلي. الغرض من عملية التصنيف هو تصنيف جميع وحدات البكسل في الصورة الرقمية إلى فئة من عدة فئات. يهدف تصنيف الصور إلى تحديد واستخراج ميزات كائن معين باستخدام صور مختلفة لنفس الكائن. هذا يساعد على تحديد أوجه التشابه بين نفس الكائن. لذلك، باستخدام هذه الميزات يمكننا بناء نظام لتصنيف الصور.

في هذا العمل سنحاول إنشاء نظام تصنيف للصور باستخدام ثلاث قواعد بيانات للصور CIFAR-10، CIFAR-100 وSTL-10 بطريقتين مختلفتين بهدف الحصول على أعلى معدل دقة ممكن.

الكلمات المفتاحية: تصنيف الصور، معالجة الصور، الرؤية الحاسوبية، التعلم الآلي، بكسل، الصورة الرقمية.

Content

Chapter 1: Image classification concepts

Chapter 2: State of the art for Image classification

Chapter 3: Experimental Results

List of tables

Chapter 2: State of the art for image classification

Chapter 3: Experimental Results

List of Figures

Chapter 1: Image classification concepts

Chapter 2: State of the art for image classification

General

Introduction

With the huge masses of digital visual information are produced nowadays, the **digital images** have become the most popular type of data which have rich content for the multiple information to extract and use them, they have become more important and unique. but computers cannot absorb and understand images as humans understand.

Giving computers the ability to see is hard because we live in a complex world (three-dimensional 3D) world and when computers try to analyze objects in 3D space, the visual sensors available like cameras usually give two dimensional (2D) images, and this projection to a lower number of dimensions incurs an enormous loss of information.

Vision allows humans to perceive and understand the world surrounding them, while computer vision aims to duplicate the effect of human vision by electronically perceiving and understanding an image.

Thanks for the increasing of volatility, necessity and applications of artificial intelligence fields like machine learning and its subsets deep learning have gained immense momentum which made this be possible more than that it has become tangible in our present. **Computer vision** tasks include methods for acquiring, processing, analyzing and understanding **digital images**, **Image classification** is perhaps the most important part of digital image analysis but it can be a complex task.

Image classification refers to the classification of images into one of the predefined categories. There are likely to be a number (n) of categories into which a particular image can be classified. When we try to manually check and classify them it might be a difficult task especially when it is huge in number, and thus it would be very helpful if we could automate this entire process using classification.

Image classification can be used in search engines such as Google or Bing image search whereby you use rich image content to query for similar stuff. Like in Google photos

where the system uses image classification to categorize your images into things like cats, dogs.

In 1st Chapter, we will introduce the different steps of image classification including pre-processing stage, features extraction stage to classification approaches based on Machine learning techniques.

In 2nd Chapter we will present the different work of classification images depending on the database used, and the result obtained.

In the last chapter, we describe the proposed model and implementation details experiments on different datasets with the different image classification techniques and their results.

Finally, at the end of the work we will try to come up with a conclusion about the obtained results, and future works that will show the potential developments of this work.

CHAPTER 1

Image classification concepts

1. Introduction

Image classification is an important step in evaluating an image and understanding its meaning which aims to effectively classify related images from large image datasets based on the extracted features of the images. These image features are usually extracted from the look, feel, and color characteristics of a query image and the images in the datasets. in this chapter, we will present some concepts about image classification.

2. Image definition

An image consists of a two-dimensional array of numbers, Images can be different for others by the way the matrix stores each pixel, the simplest type of image is binary image where each pixel is 0 or 1. The second, more complex type is gray scale, where each pixel is 256 grayscales, the most complex type of image is color. Color images are similar to grayscale except there are three bands or channels that correspond to the colors red, green, and blue (RGB) where every pixel has three values associated with it (Phillips, 1995]).

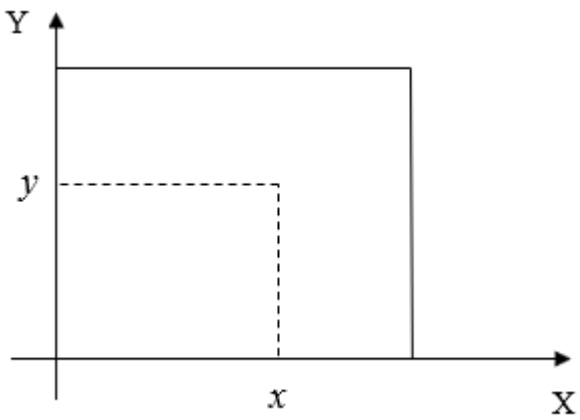

Figure 1. 1: A point of the image with coordinates (x, y).

2.1. Digital image

A digital image is a representation of an actual image as a set of numbers that can be stored and manipulated by a digital computer. In order to translate the image into numbers, it is divided into small areas called pixels (picture elements). For each pixel, the imaging device records a number, or a small set of numbers, that describe some properties of that pixel, such as its brightness (the intensity of light) or its color. The numbers are arranged in an array of rows and columns that correspond to the vertical and horizontal positions of the pixels in the image (Tyagi, 2018).

2.2. Texture

An important property of any image is the spatial structure, or arrangement, of bright and dark tones. Spatial information can be expressed in terms of spatial autocorrelation, a measure of the similarity between a pixel and its neighbors. Texture is another expression of the local spatial structure in digital images. Texture can be defined as the spatial distribution of tones of the pixels in remotely sensed images, i.e., tonal or grey level variation of an image (Christopher & al., 1998).

2.3. Image attributs

- Pixel.
- Dimension.
- Noise.
- Resolution.
- Histogram.
- Contours and textures.
- Luminance.
- Contrast.
- The weight of the image.

2.4. Different types of images

- Grayscale image.
- Monochromatic image.
- Binary image.
- Color image.

2.5. Image formats

- Bitmap images:
  - GIF format (Graphic Interchange Format).
  - JPG or JPEG (Joint Photographic Experts Group) format.
  - The PCX format (Picture Exchange Image Bitmap Zsoft).
  - PNG (Portable Network Graphics).
  - TIF/TIFF (Tag Image File Format) (Tyagi, 2018).
- Vector images:
  - The EPS (Postscript / Encapsulated Postscript) format.
  - CGM (Computer Graphics Metafile) format.

3. Image analysis issues

Giving the computer the ability to understand and recognize images is a difficult task because of:

3.1. Interpretation of images

It is a problem that a human solves unintentionally and it is the main tool for computer vision. When a person tries to comprehend an image, he relies on previous knowledge and experiences to the current observation. Where the human ability to think allows to represent the knowledge that was gathered long ago and used to solve new problems, artificial intelligence has been trying for decades to find a way to give the computer the ability to understand observations, and despite tremendous advances, the practical ability of a machine to understand observations remains very limited (Sonka & al., 1993).

3.2. Noise

Noise in digital images can come from various sources. Some are physical, linked to the nature of light and to optical artifacts, and some others are created during the conversion from electrical signal to digital data. As noise degrades the quality of an image, various models have been investigated to modelize the image noise for subsequent reduction or removal, at various steps of the image acquisition process (Julliand & al., 2016).

3.3. Too much data

With the technical advances which made processor and memory more capable than they once were (and much can be achieved with consumer level products), efficiency in problem solutions is still important and many applications remain short of real-time performance. Because nowadays, photos and videos are big. And the same thing for vision application too.

3.4. Brightness measured

Images is given by complicated image formation physics. The radiance (brightness, image intensity) depends on the irradiance (light source type, intensity and position), the observer's position, the surface local geometry, and the surface reflectance properties. which any small difference in these parameters give a new image with new values for some pixels therefore this raises the level of difficulties to extract features from it. (Sonka & al., 1993).

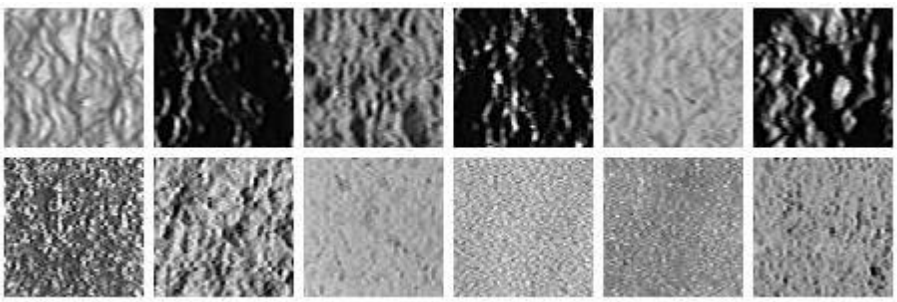

Figure 1. 2: The top row images are from the same category that look different due to variations in illumination condition and camera pose (Maboudi, 2014).

4. Machine learning and deep learning

One of the most important field of computer science is machine learning, which has achieved amazing advances in recent years, Machine learning is somewhat hard to describe succinctly, which studies algorithms and techniques to find solutions to complex problems that are difficult to program with traditional methods. the following definition will be used as a starting point, as it is sufficiently precise:

> *"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." (Mitchell, 1997).*

From a mathematical standpoint the problem can be formulated as applying a machine learning algorithm to find an unknown mathematical function. In more informal terms the program is supposed to find a mathematical connection between input and output.

The performance of an implementation of a machine learning algorithm is usually either expressed in the accuracy of the implementation, "*what percentage of the examples were classified correctly?*", or in the error rate "*what the ratio of incorrectly to correctly classified examples is?*" (Rasmus & Kristoffer, 2014).

On the other side Deep learning is a type of machine learning that works based on the structure and function of the human brain Which uses artificial neural networks to perform sophisticated computations on large amounts of data. Deep learning algorithms train machines by learning from examples (Shrestha & al., 2019).

Here's some type of Deep learning algorithms:

- Convolutional Neural Networks (CNNs)
- Long Short-Term Memory Networks (LSTMs)
- Recurrent Neural Networks (RNNs)
- Generative Adversarial Networks (GANs)
- Radial Basis Function Networks (RBFNs)
- Multilayer Perceptrons (MLPs)
- Self-Organizing Maps (SOMs)
- Deep Belief Networks (DBNs)
- Restricted Boltzmann Machines (RBMs)

For our work we will focus into two approaches, the first one based on handcraft machine learning which is textural feature oBIFs and in the second we will used Deep learning based on deep CNN.

4.1. Supervised learning

Supervised learning "or classification" consists of building a model based on a learning dataset and Labels "classes" and using it to classify new data (Zhu & Goldberg, 2009) (Soofi & al., 2017). There are several algorithms and techniques used for supervised classification such as:

- K-Nearest Neighbors (KNN)
- Bayesian classification.

- Support Vector Machine (SVM).
- Artificial Neural Networks (ANN).
- Decision tree.
- Convolutional neural networks (CNN)

4.2. Unsupervised learning

Unsupervised learning consists in providing the system with an automatic mechanism which relies on precise grouping rules to find reference classes with minimal assistance. In this case the samples are introduced in large numbers by the user without indicating their class.

There are several clustering algorithms, for example:

- Artificial Neural Networks “ANN”
- K-means “KMeans”
- Fuzzy K-Means
- Hierarchical clustering

4.3. Semi-supervised learning

Semi-supervised learning uses labeled and unlabeled dataset. It therefore falls between supervised learning which uses only labeled data and unsupervised learning which uses only unlabeled data. The use of unlabeled data, in combination with labeled data, can significantly improve the quality of learning. Another advantage comes from the fact that the data tag (labeled) requires the intervention of a human user. When the datasets become very large, this operation can be tedious. In this case, semi-supervised learning, which requires only a few labels, is of obvious and indisputable practical interest (Zhu et al., 2009).

5. Image Classification steps

All image classification systems based on ML constituted on five main steps as pre-processing, segmentation, classification (see "Figure 1.1"). In next sub-section, we will describe the different stages of an image classification system.

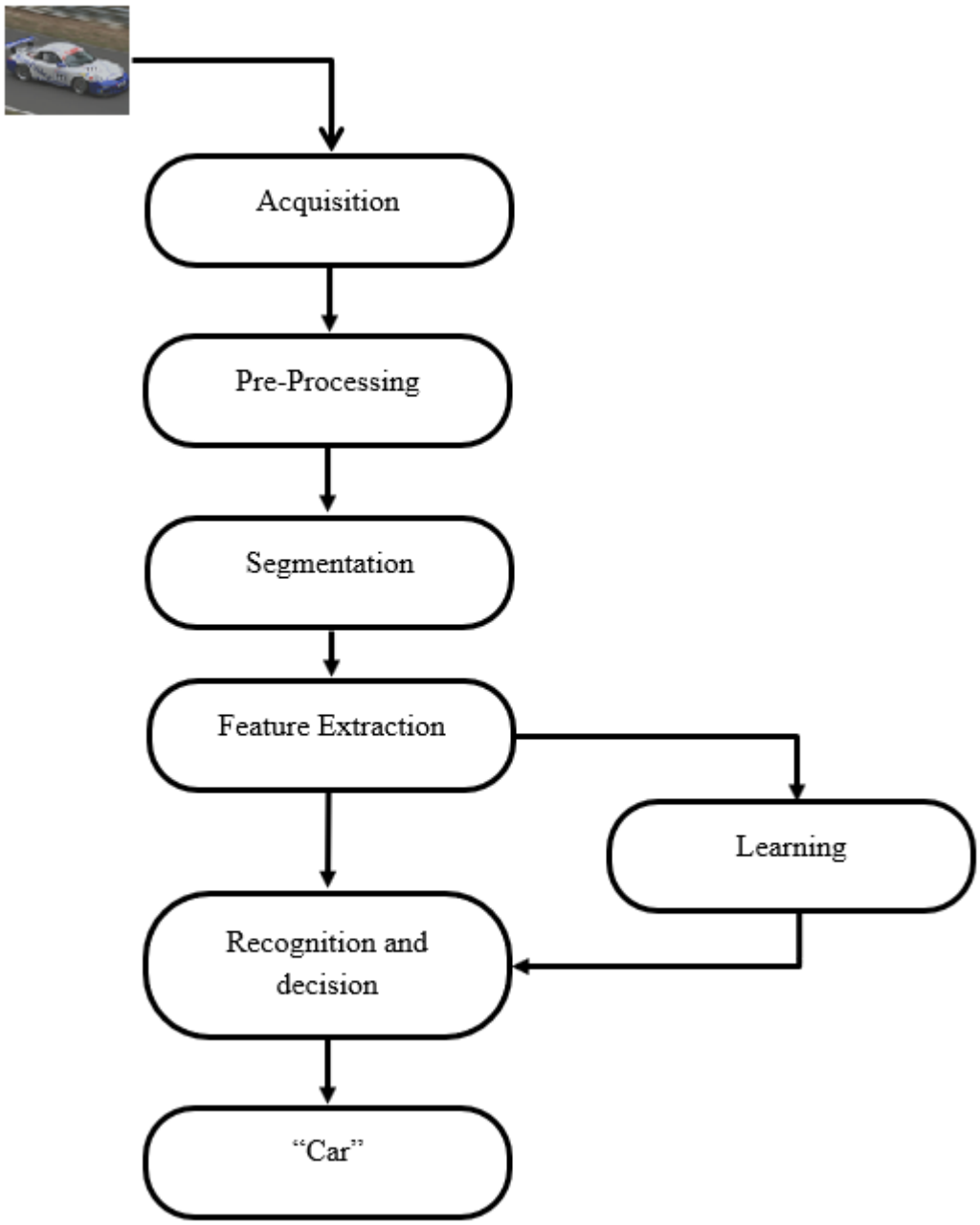

Figure 1. 3: General diagram of an image classification system.

5.1. Acquisition

Image Acquisition is to transform an optical image (Real World Data) into an array of numerical data which could be later manipulated on a computer, using physical sensors like digital cameras or scanner (Mishra & al., 2017).

5.2. Pre-processing:

The aim of this process is to improve the image data (features) by suppressing unwanted distortions and enhancement of some important image features so that the computer vision models can benefit from this improved data to work on. There're steps for image pre-processing like size normalization, slant normalization…, and Data Augmentation techniques

5.2.1. Size normalization

Normalization is the process of converting a random sized image into a standard size, some images captured by a camera vary in size, therefore, we should establish a base size for all images fed into our AI algorithms by resizing them. This is important for many neural networks and k-Nearest neighbors' algorithms.

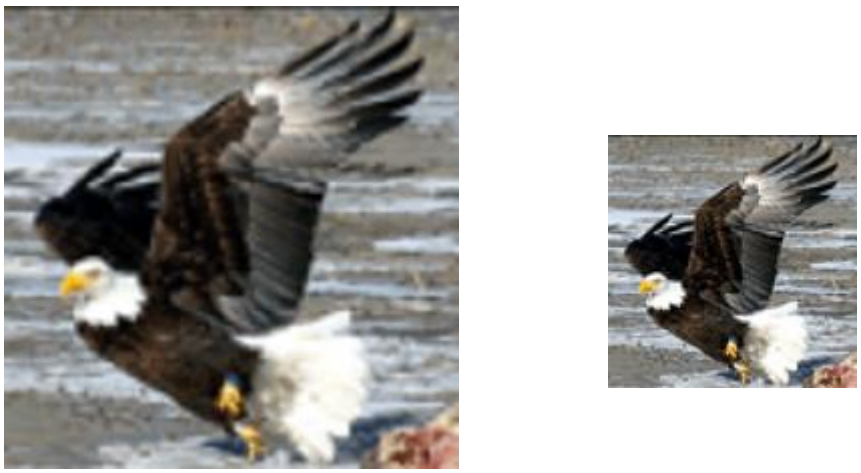

Figure 1. 4: Example for resizing image.

5.2.2. Slant normalization

Due to misalignments which is relative to the camera during capture, some meter values appear slanted in the rectangular region of interest especially in Handwriting text. The slant normalization step estimates this slant and corrects for it. To estimate the slant, the center of

mass of the dark foreground is computed in the first step. It divides the image into a left and a right pan. The centers of mass of these two parts are computed next. They define a line that corresponds to the slant estimate. To correct for the slant, a vertical sheer transformation is used that makes the line approximately horizontal. The sheering keeps the position of the center constant and shifts columns only by an integer number of pixels to avoid blaming (Behnke, 2003).

5.2.3. Gray scaling of image

The image will be converted to gray scale (range of gray shades from white to black) the computer will assign each pixel a value based on how dark it is. Grayscale image have single channel unlike the RBG images which have three channels for each pixel.

Figure 1. 5: Example for Grayscale image.

5.2.4. Gaussian Blurring

Gaussian blur or Gaussian smoothing is the result of blurring an image by a Gaussian function. It is a widely used effect in graphics software, typically to reduce image noise.

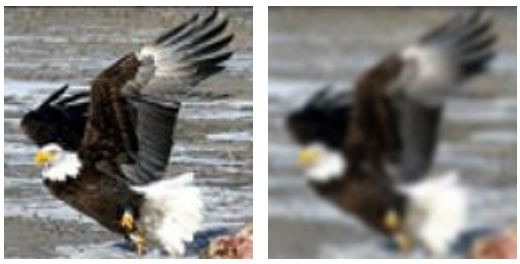

Figure 1. 6: Example for Gaussian blurring effect.

5.2.5. Histogram Equalization

Histogram equalization is another image processing technique to increase global contrast of an image using the image intensity histogram. This method needs no parameter, but it sometimes results in an unnatural looking image (Raj & al., 2015).

Figure 1. 7: Example for Histogram Equalization effect.

5.3. Data Augmentation techniques

Data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset. Training deep learning neural network models on more data can result in more skillful models, and the augmentation techniques can create variations of the images that can improve the ability of the fit models to generalize what they have learned to new images (Shorten & Khoshgoftaar, 2019).

In the following section we will describe some Data Augmentation techniques (Gu & al., 2019):

5.3.1. Rotation

A rotation augmentation rotates the image clockwise by a given number of degrees from 0 to 360. The rotation will likely rotate pixels out of the image frame and leave areas of the frame with no pixel data that must be filled in. Rotating an image might not preserve its original dimensions (depending on what angle you choose to rotate it with).

Figure 1. 8: Example for rotate an image.

5.3.2. Translation

Translation just involves moving the image along the X or Y direction (or both). This method of augmentation is very useful as most objects can be located at almost anywhere in the

image. This forces our feature extractor to look everywhere, similar to rotation, The translation will likely move pixels out of the image frame and leave areas of the frame with no pixel data which must be filled in (Shorten & Khoshgoftaar, 2019).

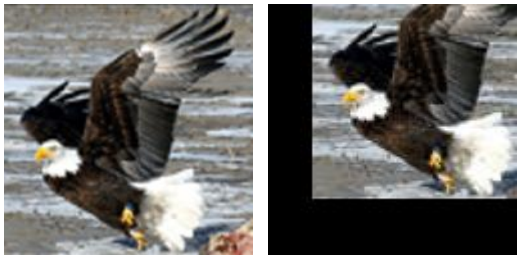

Figure 1. 9: Example for translation an image.

5.3.3. Cropping

A section of the image is selected, cropped and then resized to the original image size.

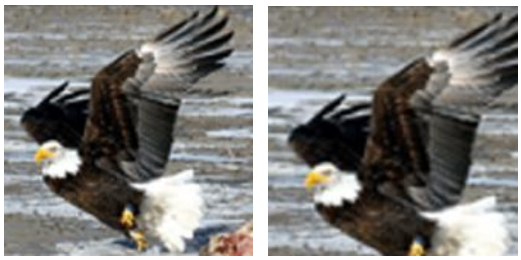

Figure 1. 10: Example for cropping an image.

5.3.4. Flipping

The image is flipped horizontally and vertically. In flipping, the pixels are rearrange while protecting features of image. Vertical flipping is not meaningful for some photos.

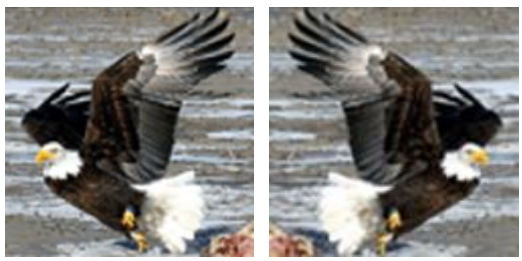

Figure 1. 11: Example for flipping an image.

5.3.5. Scaling

The image is scaled outward and inward. An object in new image can be smaller or bigger than in the original image by scaling.

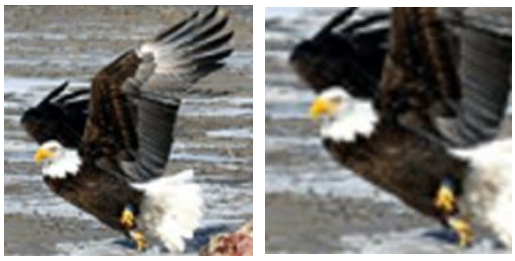

Figure 1. 12: Example for scaling an image.

5.4. Image segmentation

The process of splitting the image into many segments can be called as image segmentation. The main aim of this process is to recognize the object in an image.

Image segmentation is a most important part in the image processing, it is used almost everywhere to process the images so the model should be able to recognize what's inside the image. The segmentation splits the image into many sections or objects. The level to which the splitting the image is being carried rely on the problem is which has been solved (Glasbey & al., 1995).

5.5. Feature extraction

This is a crucial step wherein statistical or deep learning methods are used to identify the most interesting patterns of the image, features that might be unique to a particular class and that will, later on, help the model to differentiate between different classes. This process where the model learns the features from the dataset is called model training (Kunaver & Tasic, 2005).

5.6. Classification Methods

This step categorizes detected objects into predefined classes by using a suitable classification technique that compares the image patterns with the target patterns. Here some of them:

5.6.1. Neural Networks

Neural networks is a rather vague name which covers a set of calculation mechanisms initially inspired by models from the functioning of nervous systems that technicians see there as a source of inspiration for the construction of automatic systems.

A great effort has been devoted to the development of Neural Networks which present an alternative to classical architectures. This is due to their parallel structures, their Classification performance and their ability to understand nonlinear phenomena.

Mainly a Neural network only works after having learned enough knowledge about the desired outputs from given inputs; the development of these networks mainly concerns learning and the laws of modification of connection weights (Gurney, 1997).

5.6.2. K-Nearest Neighbors (KNN)

The K Nearest Neighbors Classifier It is a simple classifier based on the calculation of distance between learning examples and test examples, generally the Euclidean standard is often used as a distance measure, in each learning step, the algorithm stores the k best examples of the learning set (KNN) which are close to the test sample. This algorithm is often efficient if there are enough learning examples, but requires a very long prediction time to pass all the samples in order to find the K best solutions (Zhang & al., 2017).

5.6.3. Support Vector Machine (SVM)

SVMs are a set of supervised learning techniques intended to solve problems of binary classification and regression. SVMs are based on two ideas, the maximum margin and the kernel function.

The maximum margin is used for the problems of linear classification. It represents the distance between the separation boundary and the closest learning samples. These are the support vectors.

The kernel functions are used in the case of the problems of the non-linear classification to transform the space of representation of the input data into a space of larger dimension in which it is probable that there exist linear separators.

The SVM classifier is an algorithm that maximizes the margin between the classes of the problem to be solved and minimizes the classification error. The objective of the maximum margin is to have two classes separated by a hyperplane so that the distance from the support vectors is maximum.

In the classification task, an SVM builds the optimal hyperplane for the separation of characteristic attributes in a high dimensional space. The calculation of this hyperplane is

based on the maximization of the margin between the closest learning samples which belong to different classes (Srivastava & Bhambhu, 2010).

6. Image classification Datasets

A dataset in computer vision is a curated set of digital photographs that developers use to test, train and evaluate the performance of their algorithms. in the next part we will try to describe some of them:

6.1. CIFAR-10

“Canadian Institute for Advanced Research” is a subset of the Tiny Images dataset and consists of 60000 (32x32) color images in 10 classes. There are 50000 training images (5000 per class) and 10000 test images (1000 per class) ("CIFAR-10 & CIFAR-100 datasets", 2021).

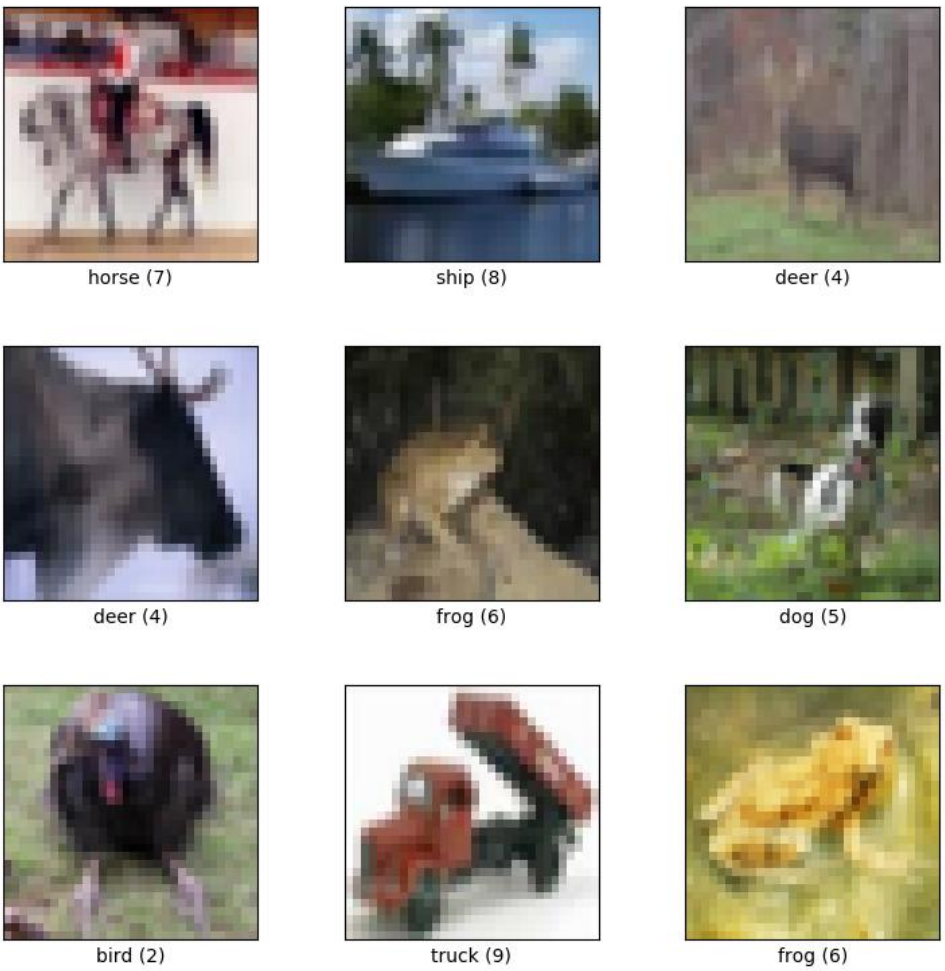

Figure 1. 13: Example of images from CIFAR-10 dataset.

6.2. CIFAR-100

“Canadian Institute for Advanced Research” is a subset of the Tiny Images dataset and consists of 60000 32x32 color images. The 100 classes in the CIFAR-100 are grouped into 20 superclasses. There are 600 images per class. Each image comes with a "fine" label (the class to which it belongs) and a "coarse" label (the superclass to which it belongs). There are 500 training images and 100 testing images per class ("CIFAR-10 and CIFAR-100 datasets", 2021).

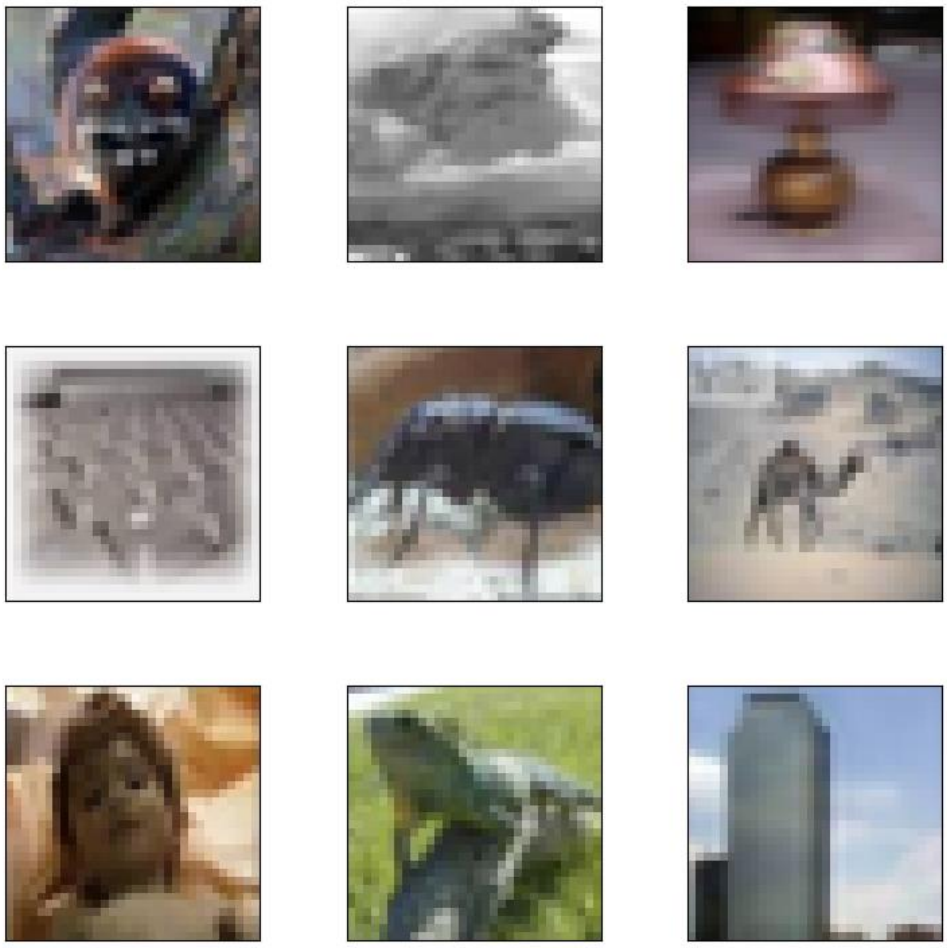

Figure 1. 14: Example of images from CIFAR-100 dataset.

6.3. STL-10

The STL-10 dataset is an image recognition dataset with (96x96) size, inspired by the CIFAR-10 dataset but with some modifications. In particular, each class has fewer labeled training examples than in CIFAR-10, but a very large set of unlabeled examples is provided to learn image models prior to supervised training. The primary challenge is to make use of the unlabeled data (which comes from a similar but different distribution from the labeled data) to build a useful prior. ("STL-10 dataset", 2021).

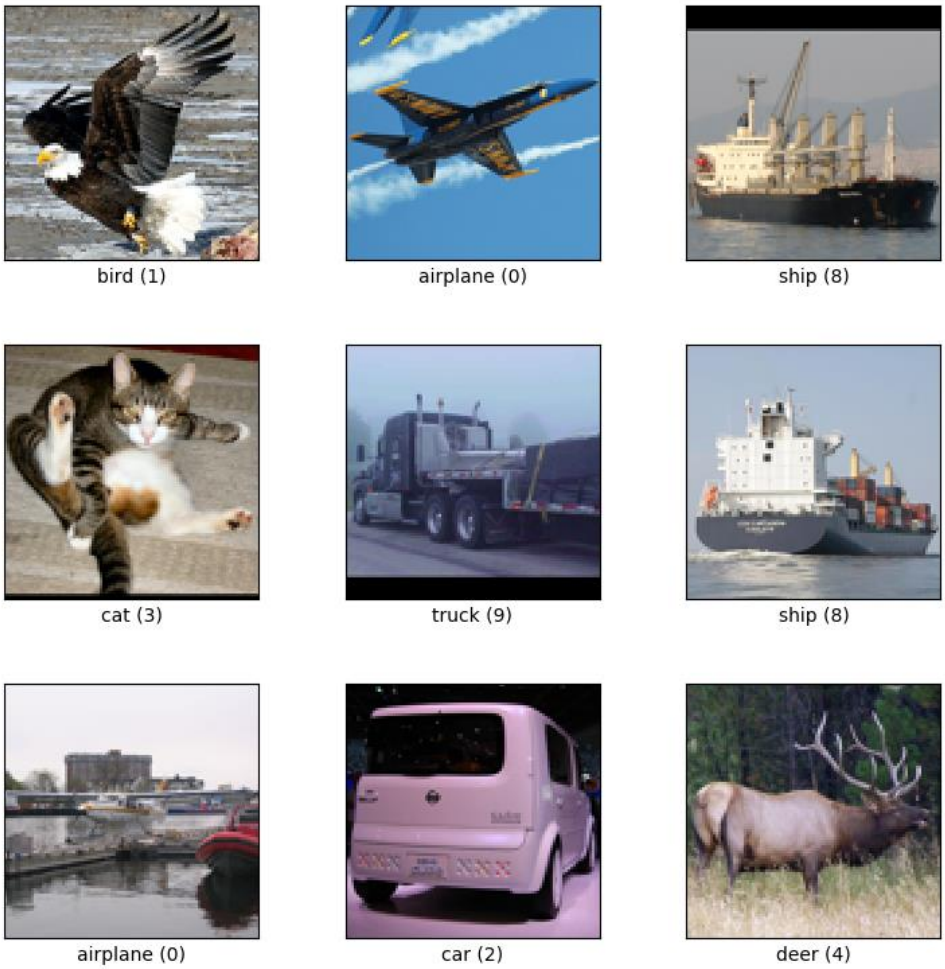

Figure 1. 15: Example of images from STL-10 dataset.

6.4. SVHN

“The Street View House Numbers**”** is a real-world image (32x32) dataset (which is obtained from house numbers in Google Street View images) for developing machine learning and object recognition algorithms with minimal requirement on data preprocessing and formatting. It can be seen as similar in flavor to MNIST, but incorporates an order of magnitude more labeled data (over 600,000 digit images) and comes from a significantly harder, unsolved, real world problem. ("The Street View House Numbers (SVHN) Dataset", 2021).

Figure 1. 16: Example of images from SVHN dataset.

6.5. MNIST

Stand for "Modified National Institute of Standards and Technology", MNIST handwritten digit classification problem is a standard dataset used in computer vision and deep learning. this dataset is effectively solved, but it can be used as the basis for learning and practicing how to develop, evaluate, and use convolutional deep learning neural networks for image classification from scratch. ("The MNIST dataset of handwritten digits", 2021).

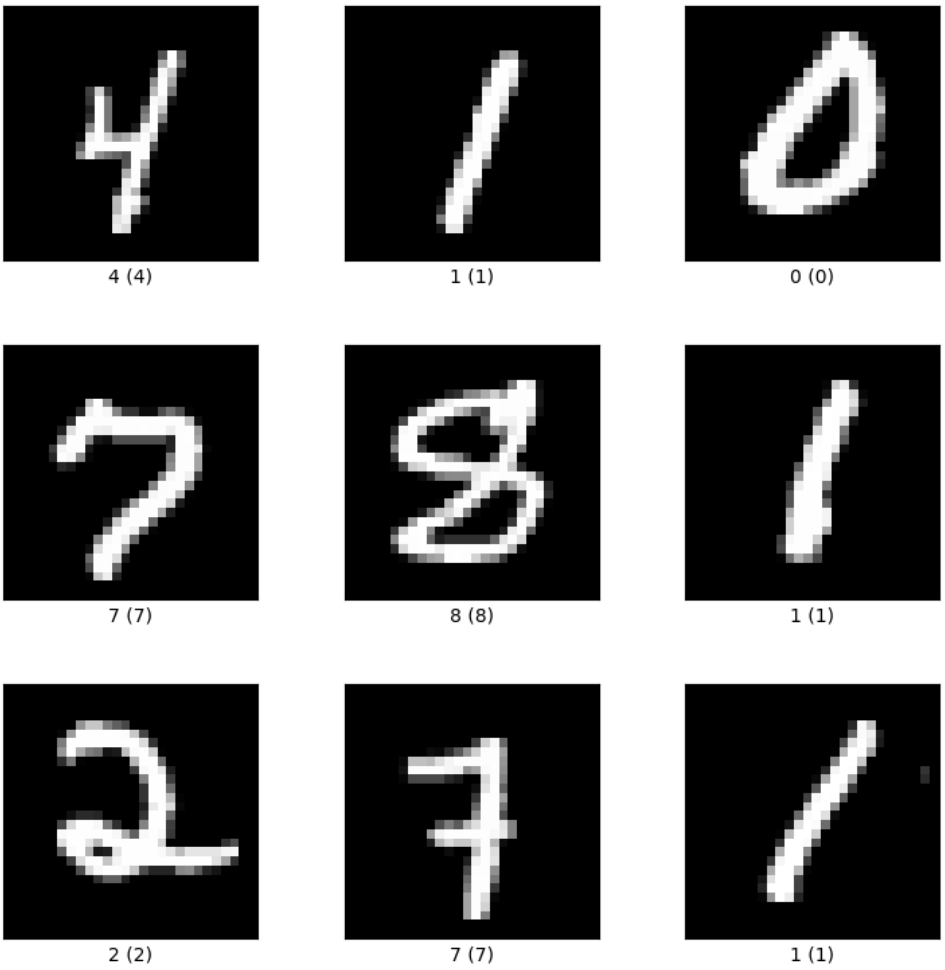

Figure 1. 17: Example of images from MNIST dataset.

6.6. Fashion-MNIST

The Fashion-MNIST dataset is proposed as a more challenging replacement dataset for the MNIST dataset. It is a dataset comprised of 60,000 small square (28×28) pixel grayscale images of items of 10 types of clothing (T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot) (Team, 2021).

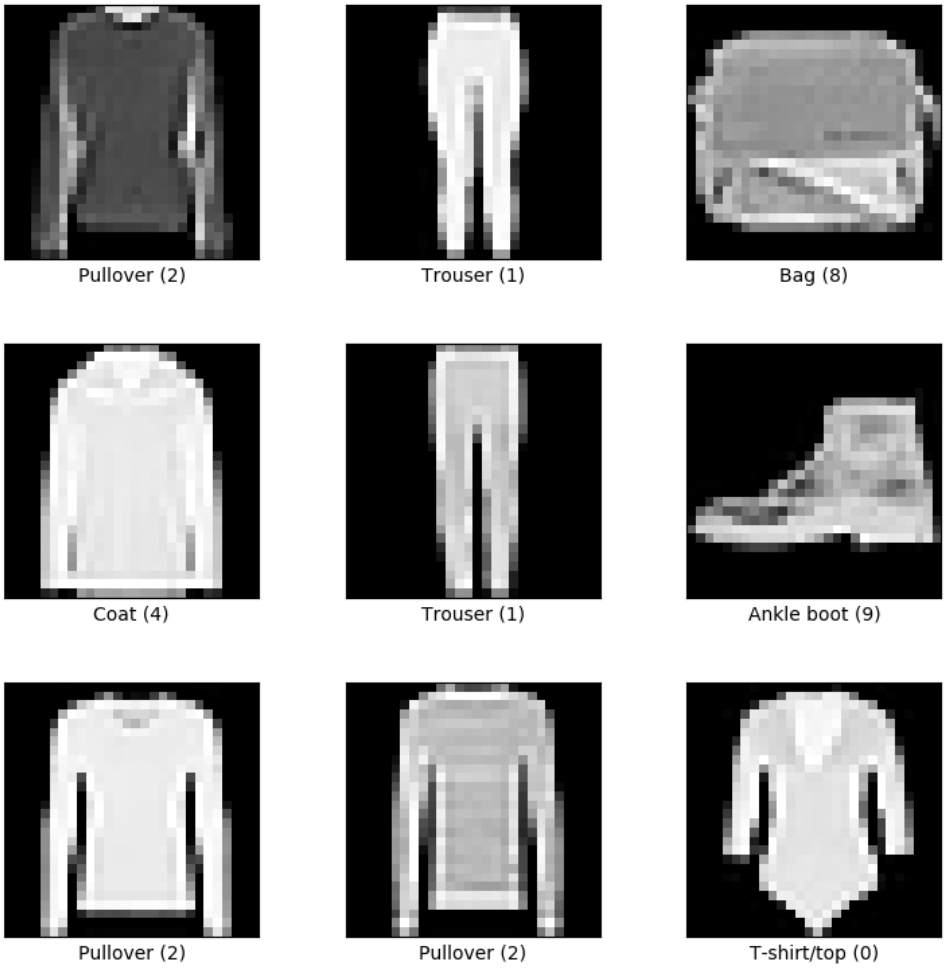

Figure 1. 18: Example of images from Fashion-MNIST dataset.

6.7. ImageNet

The ImageNet project is a large visual database designed for use in visual object recognition software research. With more than 14 million images have been hand-annotated by the project to indicate what objects are pictured and in at least one million of the images, bounding boxes are also provided. ImageNet contains more than 20,000 categories with a typical category, such as "balloon" or "strawberry" (“ImageNet dataset”, 2021).

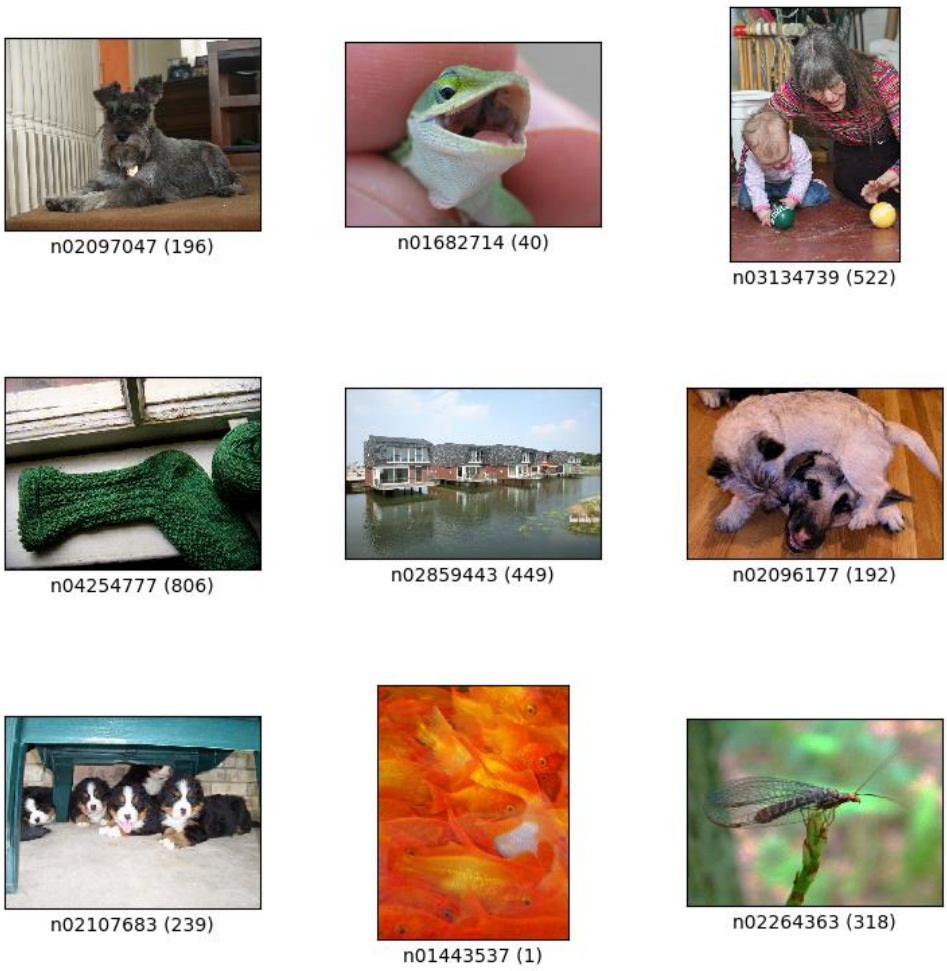

Figure 1. 19: Example of images from ImageNet dataset.

6.8. Cityscapes

Cityscapes is a large-scale database which focuses on semantic understanding of urban street scenes. Which have 30 classes grouped into 8 categories (flat surfaces, humans, vehicles, constructions, objects, nature, sky, and void). The dataset consists of around 5000 fine annotated images and 20000 coarse annotated ones. Data was captured in 50 cities during several months, daytimes, and good weather conditions. It was originally recorded as video so the frames were manually selected to have the following features: large number of dynamic objects, varying scene layout, and varying background ("Dataset Overview – Cityscapes Dataset", 2021).

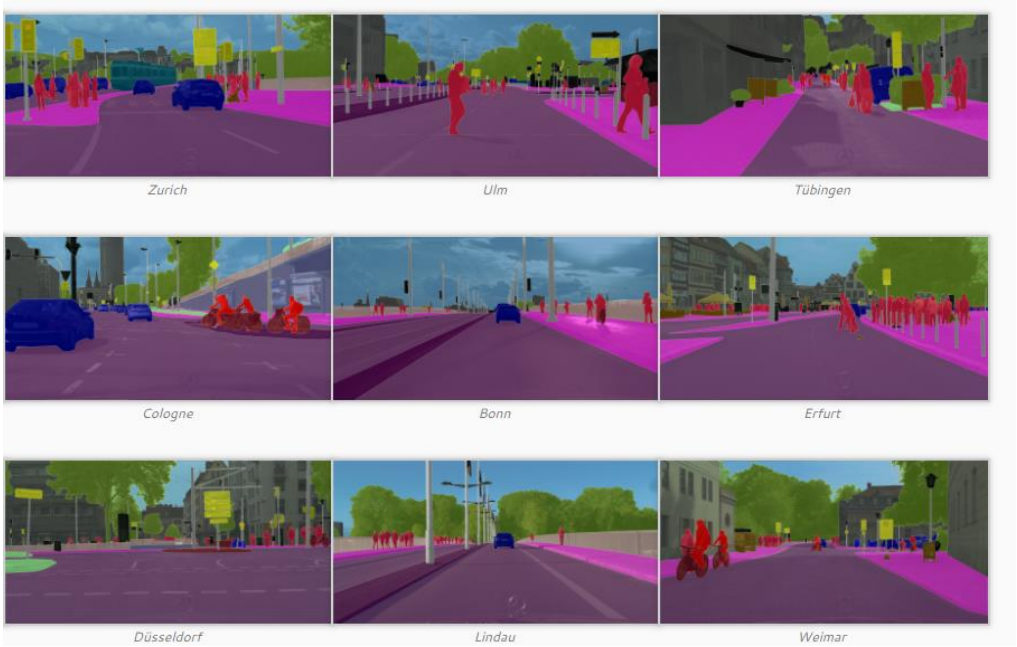

Figure 1. 20: Example of images from Cityscape's dataset.

7. Conclusion

In this chapter, we have presented in general some concepts and about images and the different stages of the image classification system, from the acquisition stage to classification stage based on machine learning and Statistical classification techniques. In the second chapter, we will address some State of art for image classification

CHAPTER 2

State of the Art for Image Classification

1. Introduction

Image classification is a fundamental task that attempts to comprehend an entire image as a whole. In recent years, several works have appeared in recent times.

In this chapter, we will present the different works of image classification depending on the method of extracting the proposed features, the database used, and the result obtained.

2. Description of various works

Several classification methods have been developed using the different recognition techniques. In what follows, we will present some works by different authors on the image classification.

2.1. Sharpness-aware minimization for efficiently improving generalization (Foret & al., 2021)

The authors present Sharpness-Aware Minimization (SAM), a novel method that improves model generalization by at the same time minimizing loss value and loss sharpness by reformulates the optimization cycle which happens during neural organization training. Rather than discovering a weight vector, which effectively minimizes the loss function for the given training set, this strategy suggests solving a different optimization problem, finding the loss function minima surrounding. Rather than utilizing Gradient-Descent to decide the function global (absolute) minima and to refresh the network weights towards that direction, their algorithm is adjusting the network, so that likewise around that point, the loss function will output minimal values.

Datasets: CIFAR-10, CIFAR-100, SVHN, Fashion-MNIST…

Here some result when SAM used in some datasets:

Task	Dataset	Metric name	Metric value (%)
Image Classification	CIFAR-10	Percentage correct	99.70
Image Classification	CIFAR-100	Percentage correct	96.08
Image Classification	Fashion-MNIST	Percentage correct	96.41
Image Classification	SVHN	Percentage error	0.99

Table 2. 1: Sharpness-aware minimization for efficiently improving generalization results on different datasets.

2.2. Fine-Tuning DARTS for Image Classification (Tanveer & al., 2020)

Fine-tuning is a proven method for improving performance of a neural network. It is a process that uses an already trained neural network on a given task and makes it perform another downstream similar task. Inspired by fine-tuning, the authors propose incorporating fixed-operations to fine-tune DARTS. using attention modules as fixed operations in their approach owing to the proven success of attention modules in improving classification accuracy.

Datasets: CIFAR-10, CIFAR-100, Fashion-MNIST, CompCars...

Task	Dataset	Metric name	Metric value (%)
Image Classification	Fashion-MNIST	Percentage error	3.09
Image Classification	CIFAR-10	Percentage correct	97.87
Image Classification	CIFAR-100	Percentage correct	84.33

Table 2. 2: Fine-Tuning DARTS for Image Classification result on different datasets.

2.3. Going deeper with Image Transformers 2021 (Hugo et al., 2021)

Transformers have recently been used for large-scale image classification and achieved high scores, shaking the long-term dominance of convolutional neural networks. However, so far, there have been few optimization studies on image transformers. In this work, the authors constructed and optimized a deeper transformer network for image classification. They specifically studied the interaction between the architecture and optimization of such specialized transformers. They made changes to the two Transformer architectures, which significantly improved the accuracy of Deep Transformer.

Datasets: CIFAR-10, CIFAR-100, ImageNet…

Here some result when this method used in some datasets:

Task	Dataset	Metric name	Metric value (%)
Image Classification	CIFAR-10	Percentage correct	99.4
Image Classification	CIFAR-100	Percentage correct	93.1

Table 2. 3: Going deeper with Image Transformers results on different datasets.

2.4. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale 2020 (Dosovitskiy et al., 2020)

Inspired by the success of Transformer scaling in NLP, the authors tried to apply the standard Transformer directly to the images with as few modifications as possible. To this end, they split the image into patches and provided linear embedding sequences of these patches as input to the Transformer. In NLP applications, image patches are handled in the same way as Tokens (words). They train image classification models in a supervised fashion. When they are trained on medium-scale datasets such as ImageNet, these models produce moderate accuracy. This seemingly frustrating result may occur: However, if the model is trained on a larger data set (14M-300M images), the picture will change. They found that large-scale training outweighed inductive bias. Their Vision Transformer achieved excellent results when pre-trained on a sufficient scale and transferred to tasks with fewer data points.

Datasets: CIFAR-10, CIFAR-100, ImageNet, Flowers-102…

Here some result when this method used in some datasets:

Task	Dataset	Metric name	Metric value (%)
Image Classification	CIFAR-10	Percentage correct	99.5
Image Classification	CIFAR-100	Percentage correct	94.55

Table 2. 4: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale results on different datasets.

2.5. FMix: Enhancing Mixed Sample Data Augmentation (Harris & al., 2020)

Mixed Sample Data Augmentation (MSDA) has received more and more attention in recent years. The authors proposed FMix, which is a type of MSDA that uses a random binary mask obtained by applying a threshold to low-frequency images sampled from Fourier space. These random masks can take a variety of shapes and can be generated for one-, two-, and three-dimensional data. FMix improves the performance of MixUp and CutMix without increasing the training time. It is suitable for multiple models across a series of data sets and problem settings. A new single model can be obtained on CIFAR-10 without external data. Finally, they showed that the result of interpolating the difference between MSDA (such as MixUp) and masking MSDA (such as FMix) is that the two can be combined to further improve performance.

Datasets: CIFAR-10, CIFAR-100, Fashion-MNIST.

Here some result when this method used in some datasets:

Task	Dataset	Metric name	Metric value (%)
Image Classification	CIFAR-10	Percentage correct	98.64
Image Classification	CIFAR-100	Percentage correct	83.95
Image Classification	Fashion-MNIST	Percentage error	3.64

Table 2. 5: FMix: Enhancing Mixed Sample Data Augmentation results on different datasets.

2.6. SpinalNet: Deep Neural Network with Gradual Input (Kabir & al., 2020)

The author proposes SpinalNet, which is created with some commonalities with the work of the human spinal cord, achieving higher accuracy with less computing resources. In the proposed SpinalNet, the structure of the hidden layer is allocated to three sectors:

- Input row.
- Intermediate row.
- output row.

The Intermediate row of SpinalNet contains some neurons. The function of input segmentation is to allow each hidden layer to receive part of the input and output of the previous layer. Therefore, the number of incoming weights in the hidden layer is significantly lower than that of traditional DNNs. Because all layers of SpinalNet directly contribute to the output line. They also studied the SpinalNet fully connected layer to several well-known DNN models, and performed traditional learning and transfer learning.

Datasets: CIFAR-10, CIFAR-100, STL-10, MNIST.

Here some result when this method used in some datasets:

Task	Dataset	Metric name	Metric value (%)
Image Classification	CIFAR-10	Percentage correct	91.98
Image Classification	CIFAR-100	Percentage correct	65.51
Image Classification	STL-10	Percentage correct	98.66
Image Classification	MNIST	Percentage correct	99.72
Image Classification	Fashion-MNIST	Percentage correct	94.68

Table 2. 6: SpinalNet: Deep Neural Network with Gradual Input results on different datasets.

2.7. EfficientNetV2: Smaller Models and Faster Training (Mingxing & Quoc, 2021)

This article introduces EfficientNetV2, which is a new family of convolutional networks, which has faster training speed and better parameter efficiency than previous models. In

order to develop this series of models, the author used a combination of training-aware neural architecture search and scaling to jointly optimize training speed and parameter efficiency. These models are searched from a search space rich in new operations (such as Fused-MBConv). Their experiments show that the training speed of the EfficientNetV2 model is much faster than the state-of-the-art model, and can be reduced by up to 6.8 times. The training speed can be further accelerated by gradually increasing the image size during training, but this usually results in a decrease in accuracy. To compensate for this decrease in accuracy, the authors suggest adaptively adjusting the regularization (dropout and data augmentation) so that they can achieve fast training and good accuracy.

Datasets: CIFAR-10, ImageNet, CIFAR-100…

Here some result when this method applied to CIFAR-10 and CIFAR-100:

Task	Dataset	Metric name	Metric value
Image Classification	CIFAR-10	Percentage correct	99.10
Image Classification	CIFAR-100	Percentage correct	92.3

Table 2. 7: EfficientNetV2: Smaller Models and Faster Training results on different datasets.

3. Comparison between subsequent work

Image classification appears to be an ever-lively research subject, which is the subject of a large number of works, thanks to its various and potential classification methods which facilitate the classification process.

In what follows we will quote the main classification systems used, the different databases and the rate obtained by each classification method as illustrated in the following table.

Dataset	Author	Title	Year	Accuracy (%)
CIFAR-10	(Pham et al)	Sharpness-Aware Minimization for Efficiently Improving Generalization	2020	99.70

	(Dosovitskiy et al)	An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale	2020	99.50
	(Dosovitskiy et al)	Going deeper with Image Transformers	2021	99.40
	(Mingxing & Quoc V)	EfficientNetV2: Smaller Models and Faster Training	2021	99.10
	(Harris & al)	FMix: Enhancing Mixed Sample Data Augmentation	2020	98.64
	(Tanveer & al)	fine-Tuning DARTS for Image Classification	2020	97.52
	(Kabir & al)	SpinalNet: Deep Neural Network with Gradual Input	2020	91.98
CIFAR-100	(Pham et al)	Sharpness-Aware Minimization for Efficiently Improving Generalization	2020	96.08
	(Dosovitskiy et al)	An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale	2020	94.55
	(Mingxing & Quoc V)	EfficientNetV2: Smaller Models and Faster Training	2021	92.3
	(Dosovitskiy et al)	Going deeper with Image Transformers	2021	93.10
	(Tanveer & al)	fine-Tuning DARTS for Image Classification	2020	84.10
	(Harris & al)	FMix: Enhancing Mixed Sample Data Augmentation	2020	83.95
	(Kabir & al)	SpinalNet: Deep Neural Network with Gradual Input	2020	65.51

Fashion-MNIST	(Tanveer & al)	fine-Tuning DARTS for Image Classification+	2020	96.91
	(Foret & al)	Sharpness-aware minimization for efficiently improving generalization	2021	96.41
	(Harris & al)	FMix: Enhancing Mixed Sample Data Augmentation	2020	96,36
	(Kabir & al)	SpinalNet: Deep Neural Network with Gradual Input	2020	94.68
MNIST	(Kabir & al)	SpinalNet: Deep Neural Network with Gradual Input	2020	99,72
STL-10	(Kabir & al)	SpinalNet: Deep Neural Network with Gradual Input	2020	98.66
SVHN	(Foret & al)	Sharpness-aware minimization for efficiently improving generalization	2021	99.01

Table 2. 8: Comparison between subsequent works result on different datasets.

4. Conclusion

In this chapter, we have made a comparative study of several previous works in the field of image classification. The main goal of this study is to help us propose a new approach by applying new methods and techniques allowing a higher accuracy rate.

Experimental Results

1. Introduction

Image classification appears to be a fascinated research subject and vast field Which attracts the interest of many researchers with various and large number of works which facilitate the classification process. but image classification still a complex process that may be affected by many factors. like determination of a suitable classification system, selection of training samples, image preprocessing, feature extraction, selection of suitable classification approaches, post-classification processing, and accuracy assessment.

2. Proposed System

The purpose of this project is to achieve an acceptable accuracy score using two different approaches, the first one is the handcraft ML based on classical oBIF and oBIF columns as feature extraction step, then KNN and SVM corresponding classifier as second step. The second approach is deep CNN as known Convolutional neural network.

The datasets that will be used to benchmark the two approaches are:

- **CIFAR-10** dataset of 60000 images (10 Classes), 50000 for training phase (5000 per class) and the remaining 10000 for testing phase (1000 per class).
- **CIFAR-100** dataset of 60000 images (100 Classes), 50000 for training phase (500 per classes) and the remaining 10000 for testing phase (100 per class).
- **STL-10** dataset of 13000 images (10 Classes), 5000 for training phase (500 per class) and remaining 8000 for test phase (800 per class).

Overall, this chapter will present an overview of the different parts of the classification and the evaluation method used for the different process. We will cover the installation and the

system requirements of the two approaches and their dependencies; as well as the framework support of programming languages. At the end of the chapter a summary is provided.

2.1. HandCraft Machine learning

In this approach, we used two steps, the first one is feature extraction based on oBIF and oBIF columns, the second one is classification step which is based on SVM and KNN as two classifiers.

2.1.1. Feature extraction step

We used in this step two textural feature oBIF and oBIF columns:

The oriented Basic Image Features (oBIFs) is an effective texture descriptor that has been successfully applied to problems like texture classification. The oriented Basic Image Features (oBIFs) is an extension to the Basic Image Features (BIFs) and involves combining the local orientation with the local symmetry information. Every location in the image is categorized into one of the seven local symmetry classes. The classification is based on the response of a bank of six Derivative-of-Gaussian (DoG) filters (up to second order) of size determined by the scale parameter (σ). A supplementary parameter ε determines if a location is to be classified as flat. To combine the local orientation information with the local symmetry information (Gattal & al., 2016).

oBIF columns: The oBIF column features are generated using various values of the scale parameter $I \in \{1, 2, 4, 8, 16\}$ while the parameter $\mathcal{E}$ is selected from a set of three small values of $\mathcal{E} \in \{0.1, 0.01, 0.001\}$. The generated feature vector is finally normalized and characterizes the writer of the document. The different steps involved in the extraction of the oBIF column features (Gattal & al. 2018).

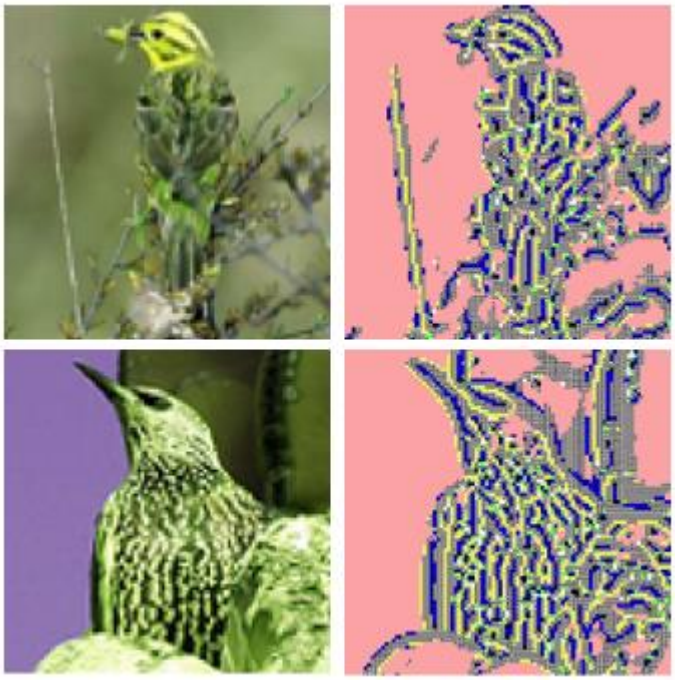

Figure 3. 1: Image with high quality features after applying oBIF.

In these images, we can distinguish between the background of the image and the objects in it, which makes the process of applying oBIF resulting with high quality features.

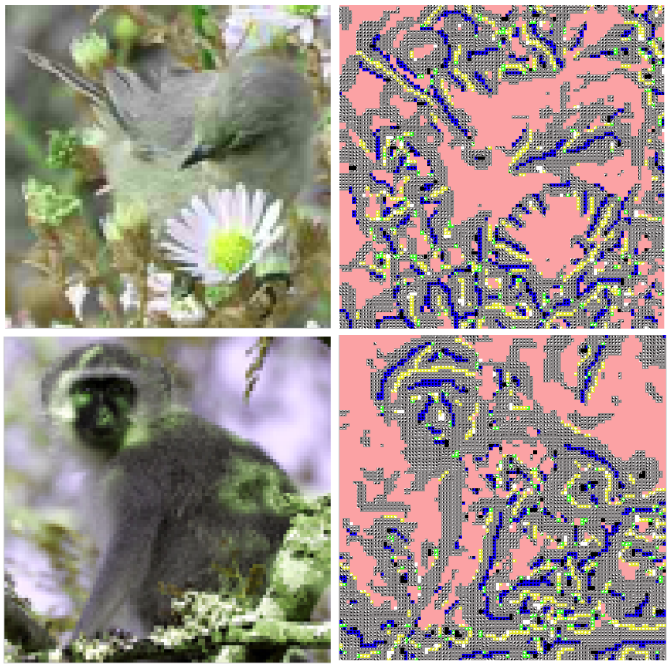

Figure 3. 2: Image with poor quality features after applying oBIF.

Unlike the first images, here we cannot distinguish between the objects and the background of the image because they blend with each other (same color, low quality images…).

2.1.2. Classification step

For the classification task we selected the (SVM) and (KNN) (based on oBIF/oBIF columns as a two classifier (Gattal & al., 2016).

Two important parameters required for training the SVM include the soft margin parameter (C) the Radial Basis Function (RBF) kernel parameter (σ). Also, for KNN we have (K) number of neighbors.

We plan to enhance the classification module by using a combination of different classifiers to arrive at the final decision about the image class.

2.2. Deep CNN

Convolutional neural networks are used to describe an architecture for applying neural networks to two-dimensional arrays like images, based on spatially localized neural input (Browne & al., 2007). This architecture has also been described as the technique of shared weights or local receptive fields (Rumelhart & al., 1986). The purpose of "CNNs" is to use spatial information between the pixels of an image. “CNNs” are particularly useful for extracting features in images. The CNN architecture includes several building blocks, such as convolution layers, pooling layers, and fully connected layers.

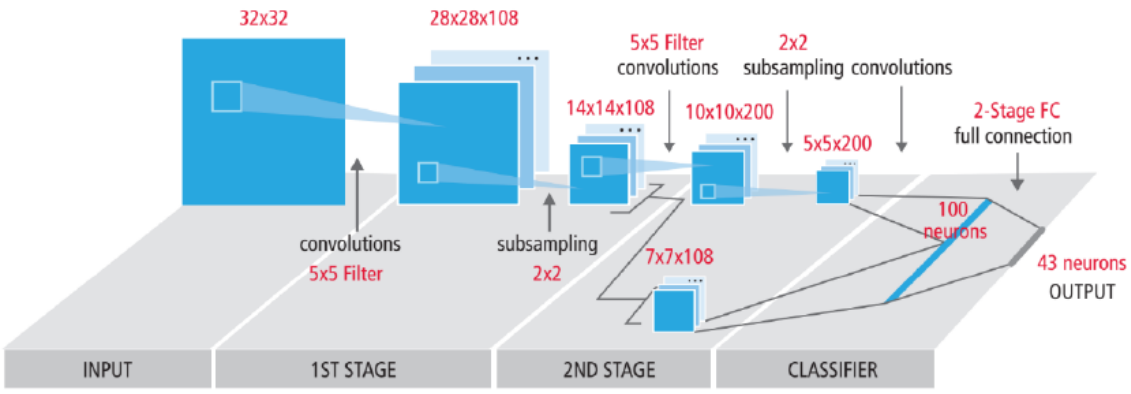

Figure 3. 3: The CNN Architecture.

By stacking several different layers in a “CNNs”, complex architectures are built to solve classification problems. The four most common types of layers are:

2.2.1. Convolutional layer

The convolution operation extracts different characteristics from the input. The first convolutional layer extracts low-level features such as edges, lines, and angles. Higher level layers extract higher level characteristics. “Figure 3.3” illustrates the convolution process used in “CNNs”.

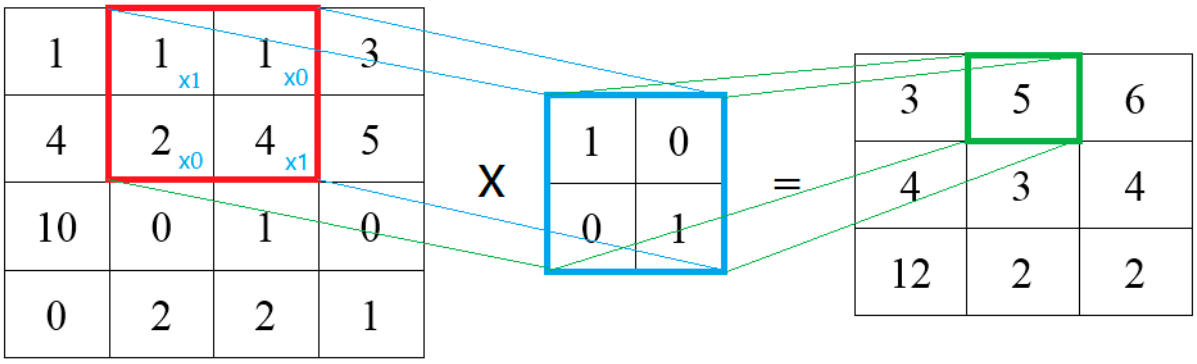

Figure 3. 4: Representation of convolution on an image of size "4x4" pixels and a convolution filter of 2x2.

At a convolution layer, the previous layer’s feature maps are convolved with learnable kernels and put through the activation function to form the output feature map. Each output map may combine convolutions with multiple input maps (Bouvrie, 2006).

2.2.2. Pooling layer

Pooling is a method of taking a large image and reducing its size while preserving the most important information it contains. here are two ways to pool (Leibe, 2014):

2.2.2.1. Max pooling

The most popular form of pooling operation is max pooling, which extracts patches from the input feature maps, outputs the maximum value in each patch, and discards all the other values. A max pooling with a filter of size 2×2 with a stride of 2 is commonly used in practice. This downsamples the in-plane dimension of feature maps by a factor of 2. Unlike height and width, the depth dimension of feature maps remains unchanged.

2.2.2.2. Average pooling

A global average pooling performs an extreme type of downsampling, where a feature map with size of height × width is downsampled into a 1×1 array by simply taking the average of all the elements in each feature map, whereas the depth of feature maps is retained. This

operation is typically applied only once before the fully connected layers. The advantages of applying global average pooling are as follows: (1) reduces the number of learnable parameters and (2) enables the CNN to accept inputs of variable size.

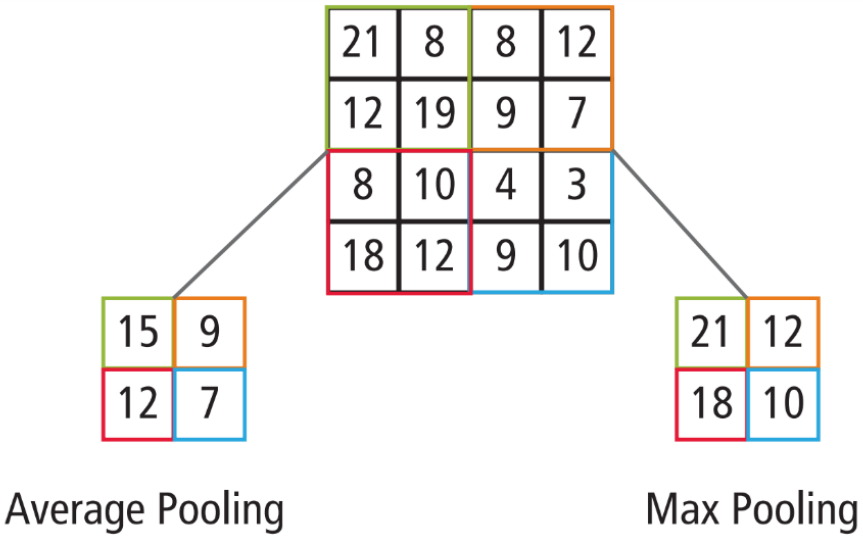

Figure 3. 5: Graphic representation of "Average pooling" and "Max pooling".

2.2.3. Nonlinear activation function

The outputs of a linear operation such as convolution are then passed through a nonlinear activation function. Although smooth nonlinear functions, such as sigmoid or hyperbolic tangent (tanh) function, were used previously because they are mathematical representations of a biological neuron behavior, the most common nonlinear activation function used presently is the rectified linear unit (ReLU), which simply computes the function:

$f(x) = max\ (0, x)$.

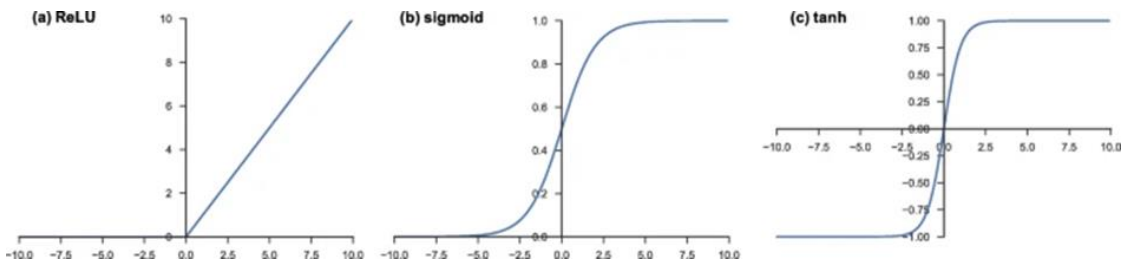

Figure 3. 6: Activation functions commonly applied to neural networks: a rectified linear unit (ReLU), b sigmoid, and c hyperbolic tangent (tanh).

For ReLU function each time there is a negative value in a pixel, it is replaced by a 0. The result of a "ReLU" layer is the same size as what is passed to it as input, with just all the negative values eliminated.

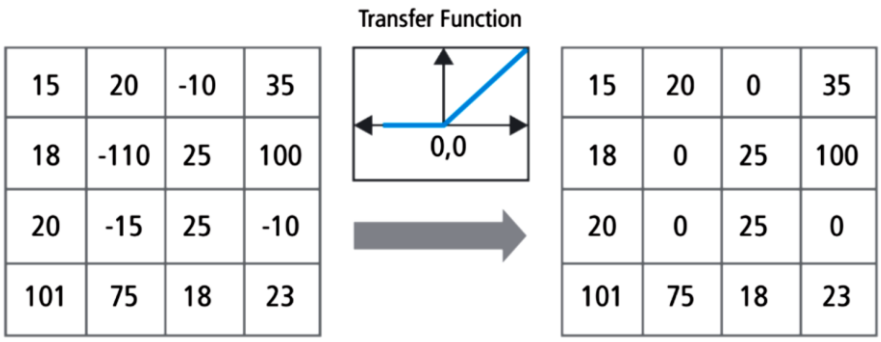

Figure 3. 7: Example for applying ReLU function.

2.2.4. Fully connected layer:

In this part the output feature maps of the final convolution or pooling layer is typically flattened which means it transformed into a one-dimensional (1D) array of numbers and connected to one or more fully connected layers, also known as dense layers, in which every input is connected to every output by a learnable weight.

Once the features extracted by the convolution layers and downsampled by the pooling layers are created, they are mapped by a subset of fully connected layers to the final outputs of the network, such as the probabilities for each class in classification tasks. The final fully connected layer typically has the same number of output nodes as the number of classes. Each fully connected layer is followed by a nonlinear function, such as ReLU (Hijazi & al., 2015).

3. Benchmarking Tests system

The System used in this part of the project:

- Operating system: 64-bit Windows 10.
- CPU: Intel Core i5-4690K @ 3.50GHz
- RAM: 8GB.
- MATLAB R2013a.

4. Development environment

4.1. Matlab

The name MATLAB is the contraction of the term "matrix laboratory", developed by the company "The MathWorks (http://www.mathworks.com)", has become a reference language for the analysis and the resolution of scientific problems.

It integrates both computing and visualization solutions and a development environment. This software is specially designed for scientific computing and vector manipulation, and the most common object in Matlab is the matrix (Bouchaib & Elhami, 2018).

4.2. Python

Python is an open source and powerful programming language, both easy to learn and rich in possibilities, the first version of which was released in 1991. It is an interpreted programming language. moreover, it is very easy to extend the existing functionalities. So, there are so called libraries which help the developer to work on particular projects. Several libraries can thus be installed to, for example, develop graphical interfaces (Bogdanchikov & al., 2013).

4.3. Google Colab

Google Colaboratory or Colab, is a simple and free Google tool that requires no configuration and runs entirely in the cloud, focus on deep learning and collaborate on data science projects. It allows you to write and execute code, save and share analyzes, and access powerful computing resources (Pessoa & al., 2018).

4.4. Kears framework

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation (Wang & al., 2020).

5. Experimentations and Results

We carried out a series of experiments to evaluate the effectiveness of the proposed system for both approaches image classification accuracy on CIFAR-10, CIFAR-100 and STL-10.

5.1. First approach: Experiment using Handcraft Machine learning

The first step for applying the feature extraction is to determine the best value Ɛ, In other words The KNN (based on oBIF/oBIF columns) applied to each dataset with different parameter for Ɛ = {1, 0.1, 0.01, 0.001} with aim to get the best value for Ɛ.

This the results obtained:

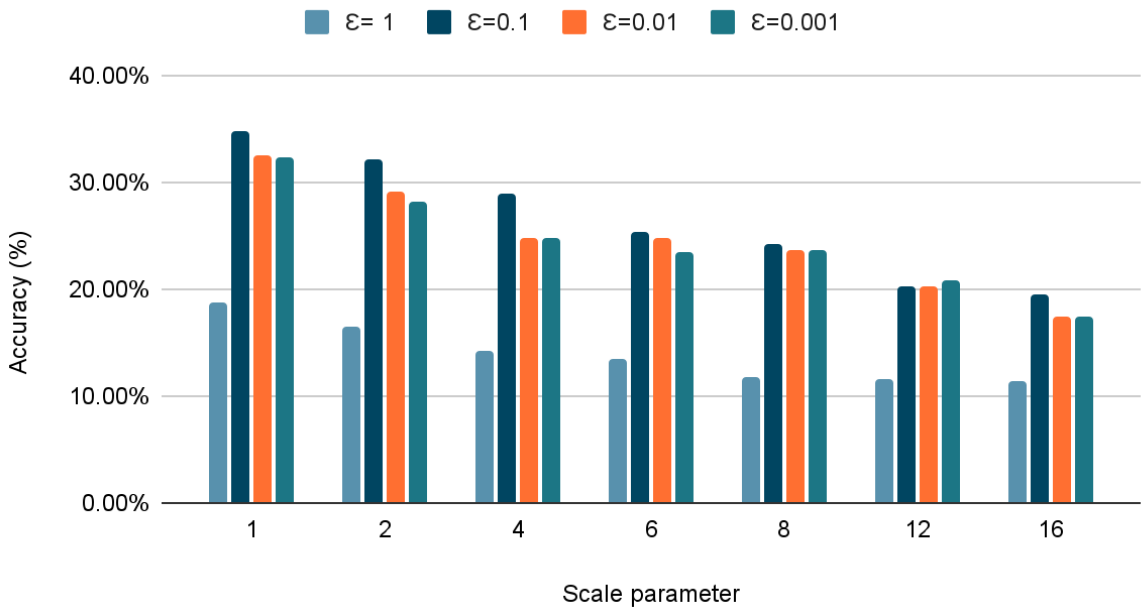

Figure 3. 8: KNN applied to CIFAR-10 Dataset.

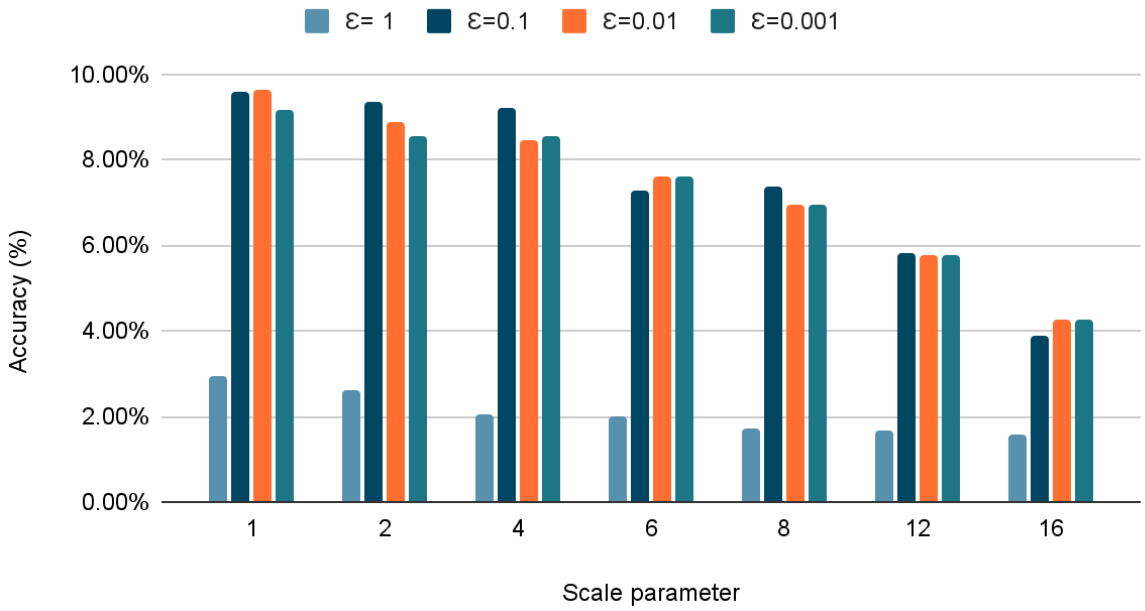

Figure 3. 9: KNN applied to CIFAR-100 Dataset.

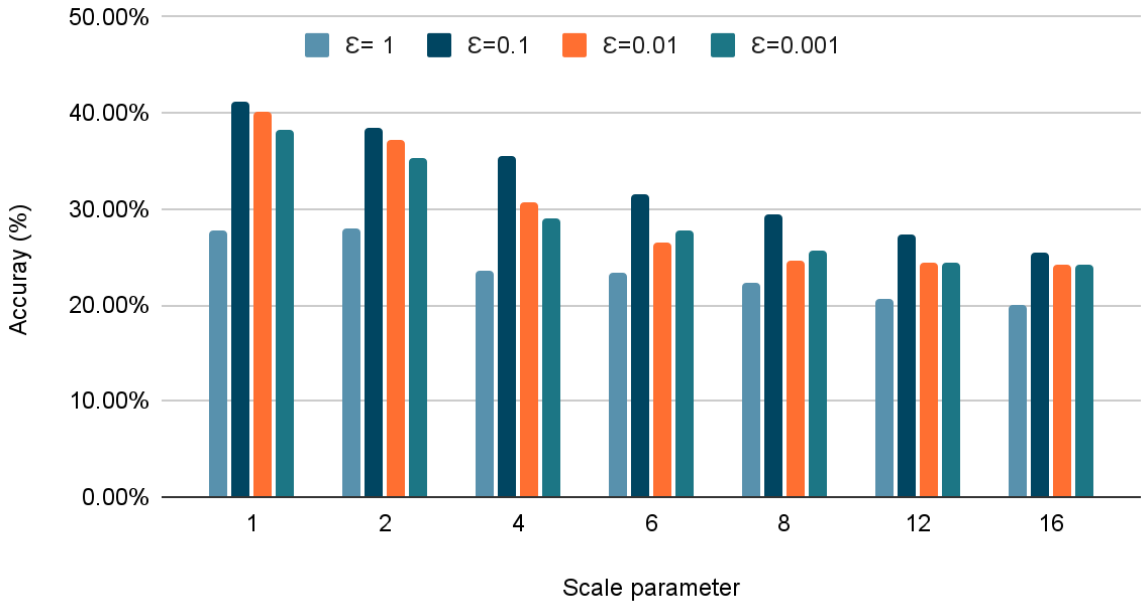

Figure 3. 10: KNN applied to STL-10 Dataset.

The results obtained after applying KNN based on oBIF, the best value for ε is 0.1. for CIFAR10, CIFAR-100, STL-10. So, we will apply KNN with ε = 0.1 the for all combination possible on CIFAR-10, CIFAR-100, STL-10.

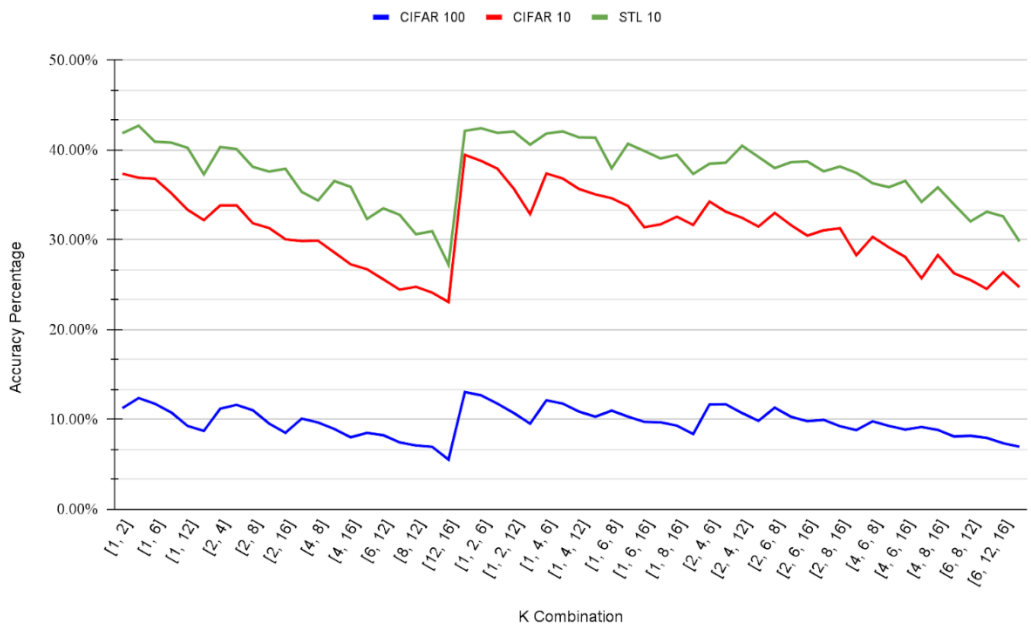

Figure 3. 11: Classifying accuracy using KNN based on oBIF for CIFAR-10, CIFAR-100, STL-10 Datasets.

Now the same thing but with oBIF Columns instead of oBIF:

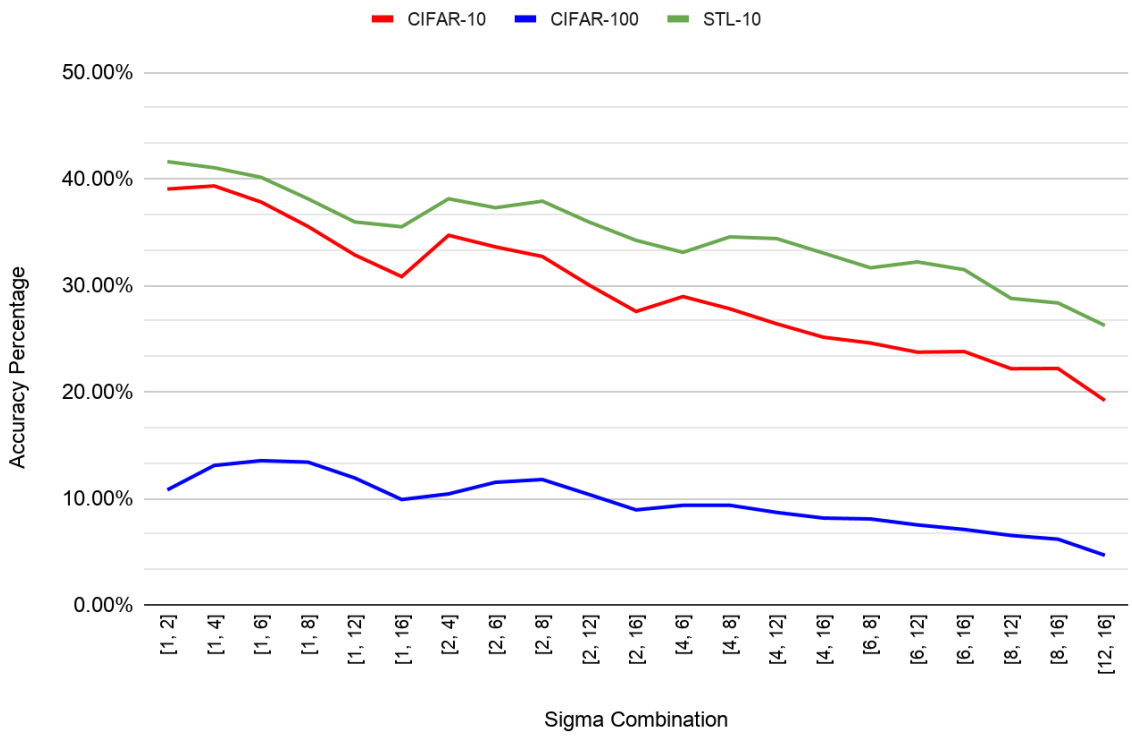

Figure 3. 12: Classifying accuracy using KNN based on oBIF Columns for CIFAR-10, CIFAR-100, STL-10 Datasets.

After that we will choose best combination based on top accuracy obtained, Next, we applied SVM based on that combination, The results can be summarized in the following table.

Note: we failed to get results from CIFAR-10 and CIFAR-100 when we applied SVM due to the limitation of hardware.

Datasets	**Feature parameter**	**Size**	**Classifier parameter**	**Accuracy (%)**				
				KNN	**K**	**SVM**	**C**	σ
STL-10	oBIF	96x96	Ɛ = 0.1, σ = [1,4]	42.6875	10	53.60	10	10
	oBIF		Ɛ = 0.1, σ = [1, 2, 6]	42.40	10	55.10	10	11
	oBIF Columns		Ɛ = 0.1, σ_1 = 1, σ_2 = 2	41.6125	9	53.95	10	13
	oBIF Columns		Ɛ = 0.1, σ1 = 1, σ2 = 4	41.0375	6	54.75	10	13
CIFAR-10	oBIF	32x32	Ɛ = 0.1, σ = [1, 2, 4]	39.4400	15	/	/	/
	oBIF		Ɛ = 0.1, σ = [1, 2, 6]	38.7700	11	/	/	/
	oBIF Columns		Ɛ = 0.1, σ_1 = 1, σ_2 = 4	39.3300	10	/	/	/
	oBIF Columns		Ɛ = 0.1, σ1 = 1, σ2 = 2	39.0400	12	/	/	/
CIFAR-100	oBIF	32x32	Ɛ = 0.1, σ = [1, 2, 4]	13.0100	6	/	/	/
	oBIF		Ɛ = 0.1, σ = [1, 2, 6]	12.6400	6	/	/	/
	oBIF Columns		Ɛ = 0.1, σ_1 = 1, σ_2 = 6	13.5500	4	/	/	/

	oBIF Columns		Ɛ = 0.1, σ1 = 1, σ2 = 8	13.4100	4	/	/	/

Table 3. 1: All results obtained when we applied KNN (oBIF/oBIF columns) to CIFAR-10, CIFAR-100 and STL-10.

5.2. Discussion

As shown in Table 3.1 above, we note that the highest accuracy rate for:

STL-10: when we applied KNN (based on oBIF) is 42.68% where the distance K= 10 with parameter Ɛ= 0.1 and the combination is [1,4]. And for KNN (oBIF columns) the result was 41.61% with parameter Ɛ = 0.1 and combination σ = [1,2,6].

In the other hand SVM result was better with 55.10% where C = 10 & σ = 13, the combination was $\sigma_1 = 1$, $\sigma_2 = 2$.

CIFAR-10: the KNN (based on oBIF) result is 39.44% where the distance K= 15 with parameter Ɛ= 0.1 and the combination is [1, 2, 4]. for KNN (oBIF columns) the result was 39.33% with parameter Ɛ = 0.1 and with combination σ1 = 1, σ2 = 4.

CIFAR-100: The task of classifying this dataset was difficult because it contained 100 classes. The accuracy rate obtained when we applied KNN were low, reaching 13.01% for KNN (based on oBIF) with Ɛ = 0.1, combination σ = [1, 2, 4]. And 13.55% for KNN (based on oBIF columns) where Ɛ = 0.1, and combination $\sigma_1 = 1$, $\sigma_2 = 6$.

Unfortunately, we did not get results for SVM for both CIFAR-10 and CIFAR-100 due to the limitation of hardware.

5.3. Second approach: Experiment using Deep CNN

The general architecture for our model based on (Visual Geometry Group) VGG model (Simonyan & Zisserman, 2014). because this model structure of the architecture is easy to understand and implement. Where the architecture involves stacking convolutional layers with small 3×3 filters followed by a max pooling layer. which form a block, and these blocks can be repeated where the number of filters in each block is increased with the depth of the network such as {32, 64, 128} for the first three blocks of the model. Padding is used on the convolutional layers to ensure the height and width of the output feature maps matches the inputs. Each layer will use the ReLU activation function. Lastly, we can explore this

architecture on the CIFAR-10, CIFAR-100 and STL-10 with different filters numbers combination.

5.3.1. Three block CNN

The input images are of size (32x32) for both CIFAR-10 and CIFAR-100, but for STL-10 is (96x96).

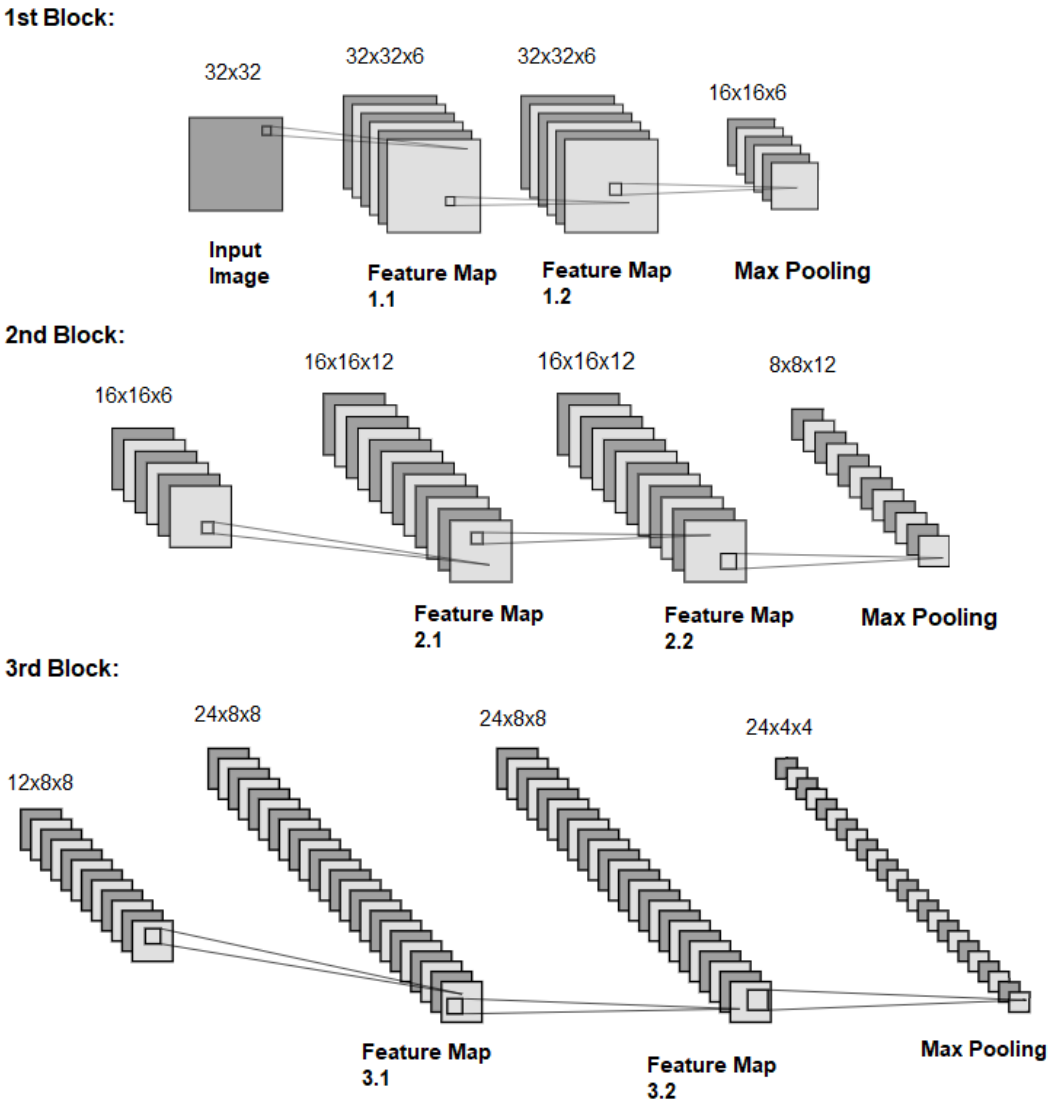

Figure 3. 13: The proposed architecture of (CNNs). FM 1.1/1.2 = 6, FM 2.1/2.2 = 12, MF 3.1/3.2 = 64.

The convolutional layers have:

- filters of size (3 x 3) with these parameters
- Padding = 1.
- Stride = 1.

- Activation function is ReLU

For max Pooling layer:

- Kernal (2x2)
- Stride = 2.

for Fully connected layer (Classification step):

- CIFAR-10 with 10 Neurons.
- CIFAR-100 with 100 Neurons.
- STL-10 with 10 Neurons.
- Activation function for last layer is SoftMax

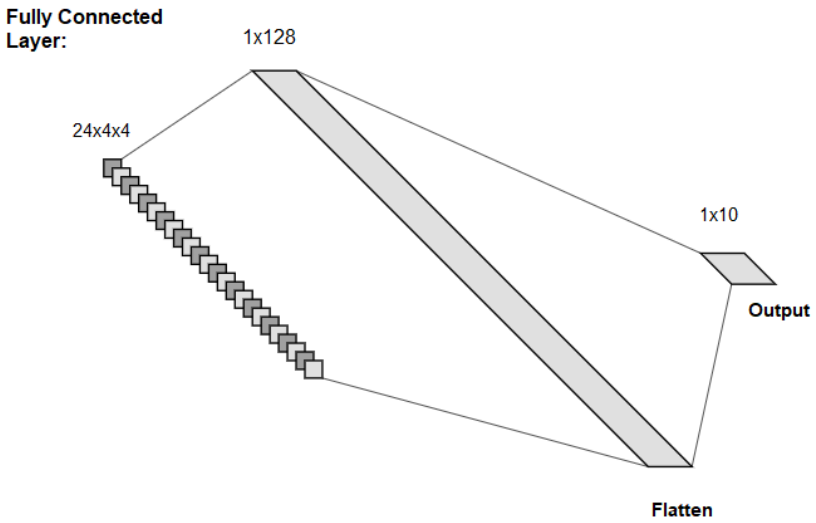

Figure 3. 14: Fully connected layer architecture for our model.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_6 (Conv2D)            (None, 32, 32, 32)        896
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 32, 32, 32)        9248
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 16, 16, 64)        18496
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 16, 16, 64)        36928
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 8, 8, 64)          0
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 8, 8, 128)         73856
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 8, 8, 128)         147584
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 4, 4, 128)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 2048)              0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               262272
_________________________________________________________________
dense_3 (Dense)              (None, 10)                1290
=================================================================
Total params: 550,570
Trainable params: 550,570
Non-trainable params: 0
```

Figure 3. 15: Summary for the proposed three block CNN architecture. FM 1.1/1.2 = 6, FM 2.1/2.2 = 12, MF 3.1/3.2 = 64.

Here's the accuracy rate with all patterns:

Dataset	**Image Size**	**Feature Map (1.1/1.2)**	**Feature Map (2.1/2.2)**	**Feature Map (3.1/3.2)**	**Epoch**	**Accuracy (%)**
CIFAR-10	32x32	4	8	16	24	62.97
		6	12	24	13	66.51
		8	16	32	14	68.09
		10	20	40	17	70.77
		12	24	48	10	72.32
		14	28	56	8	72.73
		16	32	64	17	73.25
		32	64	128	11	78.02

CIFAR-100	32x32	4	8	16	36	30.02
		6	12	24	19	34.29
		8	16	32	16	35.60
		10	20	40	13	36.61
		12	24	48	13	37.27
		14	28	56	12	38.08
		16	32	64	11	39.50
		32	64	128	6	42.57
STL-10	96x96	4	8	16	11	52.82
		6	12	24	5	54.76
		8	16	32	6	52.65
		10	20	40	42	55.89
		12	24	48	49	55.61
		14	28	56	8	56.85
		16	32	64	50	58.55
		32	64	128	49	55.11

Table 3. 2: Classifying accuracy using the proposed CNN for CIFAR-10, CIFAR-100 and STL-10 Datasets.

5.3.2. Discussion

In Table 3.2 above, we can see the highest accuracy rate that we got for:

- **CIFAR-10** dataset accuracy rate is 78.02% with FM1.1/1.2 = 32, FM2.1/2.2 = 64, FM3.1/3.2 = 128 with 11 epochs.
- **CIFAR-100** dataset accuracy rate is 42.57% with FM1.1/1.2 = 32, FM2.1/2.2 = 64, FM3.1/3.2 = 128 with 6 epochs.
- **STL-10** dataset accuracy rate is 58.55% where FM1.1/1.2 = 16, FM2.1/2.2 = 32, FM3.1/3.2 = 64 with 50 epochs.

On the other hand, the lowest accuracy rate reached for:

- **CIFAR-10** dataset accuracy rate is 62.97% with FM1.1/1.2 = 4, FM2.1/2.2 = 8, FM3.1/3.2 = 16 with 24 epochs.
- **CIFAR-100** dataset accuracy rate is 30.02% with FM1.1/1.2 = 4, FM2.1/2.2 = 8, FM3.1/3.2 = 16 with 36 epochs.

- **STL-10** dataset accuracy rate is 52.82% where FM1.1/1.2 = 4, FM2.1/2.2 = 8, FM3.1/3.2 = 16 with 11 epochs.

The results of the model on the test dataset showed an improvement in classification accuracy compared to first approach.

5.3.3. Adding Dropout for the proposed CNNs architecture

Now we will try to improve the accuracy rate by adding Dropout which is a simple technique that will randomly drop nodes out of the network. It has a regularizing effect as the remaining nodes must adapt to pick-up the slack of the removed nodes (Srivastava & al., 2014).

This technique will be used between the blocks and also in last layer which is fully connected layer. We choose (20%) as Dropout value.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 32, 32, 6)         168
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 32, 32, 6)         330
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 16, 16, 6)         0
_________________________________________________________________
dropout (Dropout)            (None, 16, 16, 6)         0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 16, 16, 12)        660
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 16, 16, 12)        1308
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 8, 8, 12)          0
_________________________________________________________________
dropout_1 (Dropout)          (None, 8, 8, 12)          0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 8, 8, 24)          2616
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 8, 8, 24)          5208
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 4, 4, 24)          0
_________________________________________________________________
dropout_2 (Dropout)          (None, 4, 4, 24)          0
_________________________________________________________________
flatten (Flatten)            (None, 384)               0
_________________________________________________________________
dense (Dense)                (None, 128)               49280
_________________________________________________________________
dropout_3 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 10)                1290
=================================================================
Total params: 60,860
Trainable params: 60,860
Non-trainable params: 0
```

Figure 3. 16: Summary for the proposed three block CNN architecture with Dropout. FM 1.1/1.2 = 6, FM 2.1/2.2 = 12, MF 3.1/3.2 = 64.

Here's the new accuracy rate after we applied Dropout.

Dataset	Image Size	Feature Map (1.1/1.2)	Feature Map (2.1/2.2)	Feature Map (3.1/3.2)	Epoch	Accuracy (%)
CIFAR-10	32x32	4	8	16	48	64.83
		6	12	24	49	71.76
		8	16	32	49	72.89
		10	20	40	42	76.97
		12	24	48	47	79.60
		14	28	56	38	79.13
		16	32	64	41	80.06
		32	64	128	48	82.01
CIFAR-100	32x32	4	8	16	48	28.79
		6	12	24	49	33.41
		8	16	32	42	38.61
		10	20	40	48	41.97
		12	24	48	50	41.24
		14	28	56	43	44.68
		16	32	64	48	45.34
		32	64	128	50	48.25
STL-10	96x96	4	8	16	41	51.56
		6	12	24	18	55.01
		8	16	32	33	56.98
		10	20	40	39	56.41
		12	24	48	28	57.91
		14	28	56	48	59.40
		16	32	64	41	58.61
		32	64	128	49	59.81

Table 3. 3: Classifying accuracy using the proposed CNN with Dropout for CIFAR-10, CIFAR-100 and STL-10 Datasets.

5.3.4. Discussion

In this case, when we add Dropout function, the accuracy rate that we get for:

- **CIFAR-10** dataset accuracy rate is 82.01% with FM1.1/1.2 = 32, FM2.1/2.2 = 64, FM3.1/3.2 = 128 with 48 epochs.
- **CIFAR-100** dataset accuracy rate is 48.25% with FM1.1/1.2 = 32, FM2.1/2.2 = 64, FM3.1/3.2 = 128 with 50 epochs.
- **STL-10** dataset accuracy rate is 59.81% where FM1.1/1.2 = 32, FM2.1/2.2 = 64, FM3.1/3.2 = 128 with 49 epochs.

On the other hand, the lowest accuracy rate reached for:

- **CIFAR-10** dataset accuracy rate is 64.83% with FM1.1/1.2 = 4, FM2.1/2.2 = 8, FM3.1/3.2 = 16 with 48 epochs.
- **CIFAR-100** dataset accuracy rate is 28.79% with FM1.1/1.2 = 4, FM2.1/2.2 = 8, FM3.1/3.2 = 16 with 48 epochs.
- **STL-10** dataset accuracy rate is 51.56% where FM1.1/1.2 = 4, FM2.1/2.2 = 8, FM3.1/3.2 = 16 with 41 epochs.

Overall, the Dropout technique improved the accuracy rate by 2-7%. Which means we can achieve a higher accuracy rate with more parameters.

6. Future scope for both approaches

In all cases, the two approaches were able to learn from the training dataset, showing an improvement on the training dataset. This is a good sign, as it shows that the problem is learnable and that the two approaches have sufficient capacity to learn the problem.

The results of the first approaches on the test dataset showed an improvement in classification accuracy with each different parameter. It is possible that this trend would continue if we tried different Feature extraction techniques, and this might make an interesting extension.

On the other side the results of the second approach which is Deep CNN on the test datasets were great especially for CIFAR-10. Overall, the second approach showed an improvement in classification accuracy with each increase in the depth of the model. Also, we got an improvement when we added Dropout which means we can get a higher accuracy rate with tweaking on CNN parameters like (Change Kernel Sizes, Activation Functions, Change the

architecture of CNNs, Increasing the number of training epochs to give the model enough space, Change dropout from 20% to 25% or 30%... etc.).

also, it may be useful to investigate techniques that slow down the convergence (rate of learning) of the model. This may include techniques such as data augmentation as well as learning rate schedules, changes to the batch size, and perhaps more.

7. Conclusion

In this chapter, we have presented the tools, the development environment and the feature extraction techniques also the classifiers that we used in this work and the results. all of this has been devoted mainly to the evaluation of the proposed methods for the image classification for three different datasets. The main purpose of this evaluation is to help us choose the best combination of KNN and SVM also the best pattern for CNNs which will take the highest accuracy rate.

General Conclusion

Image classification is an important task in the field of computer vision, object recognition and machine learning. Although the capacities of the activities carried out in the field of image classification are numerous, no method is considered completely weak, every new work tries to improve the scores for better results.

The objective of this work devoted to study classification method for images on three Datasets using two approaches one based on Machine learning where we used oBIF and oBIF columns as feature extraction step then for classification we choose KNN & SVM. For the second approach we tried to scratch deep learning field by using Deep CNN. this choice is justified by the simplicity and efficiency of those methods.

During this research, we carried out a series of experiments to evaluate the effectiveness of the proposed system for image classification. We investigated the oriented Basic Image Features (oBIFs) for this purpose. The features are extracted using different parameter settings, while the classification is carried out using SVM and KNN. The system evaluated on three different datasets CIFAR-10, CIFAR-100 & STL-10. First, the datasets comprised 60000 for CIFAR-10, and 60000 also for CIFAR-100, And 13000 for STL-10.

The result obtained during the test phase confirms the effectiveness of our two approaches. Though, our work is only in its initial version, we can say that this work remains open for comparison and/or hybridization work with other classification methods.

Finally, the conclusions of this work are that the whole system needs to be efficient. The image classification system needs to be accurate. And this work set to the initial point for that, despite the results obtained, increase of performance and improvements can be considered with:

- Applying various data augmentation techniques to increase the size of the training set.
- Appling various filters to the dataset to make feature extraction techniques more efficient.
- Applying multiple features extraction techniques on image and compare it with original image.
- Check other parameters.
- Using other technique like transfer learning.
- Improve the system by combining the classifier with other or more classifiers.
- Using another classification method.
- Test with other datasets which have a greater number of training set, to include more variations in images. so that generalization can be more efficient.

References

[1] Phillips, D. (1995). *Image processing in C*. BPB Publications. pp 7-8.

[2] Tyagi, V. (2018). Introduction to Digital Image Processing. *Understanding Digital Image Processing*, pp 1-12. doi:10.1201/9781315123905-1

[3] Christopher, F & Sebastian J. (1998). Scale and texture in digital image classification. pp 2-2. doi:10.33915/etd.93

[4] Tyagi, V. (2018). Introduction to Digital Image Processing. *Understanding Digital Image Processing*. pp 8-9. doi:10.1201/9781315123905-1

[5] Sonka, M., Vaclav H & Roger B. (1993). Image processing, analysis, and machine vision. *International Thomson, London*. p4.

[6] Julliand, T., Nozick, V., & Talbot, H. (2015, October). Image noise and digital image forensics. In *International Workshop on Digital Watermarking* (pp. 3-17). Springer, Cham.

[7] Sonka, M., Vaclav H & Roger B. (1993) Image processing, analysis, and machine vision. *International Thomson, london*. p4.

[8] Maboudi Afkham, H. (2014). *Improving Image Classification Performance using Joint Feature Selection* (Doctoral dissertation, KTH Royal Institute of Technology). p9.

[9] Mitchell, T. M. (1997). Machine learning.

[10] Rasmus, A & Kristoffer, H. (2014). Image classification, deep learning and convolutional neural networks: a comparative study of machine learning frameworks. p5.

[11] Shrestha, A., & Mahmood, A. (2019). Review of deep learning algorithms and architectures. *IEEE Access*, *7*, 53040-53065.

[12] Zhu, X., & Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, *3*(1), p3-4.

[13] Soofi, A. A., & Awan, A. (2017). Classification techniques in machine learning: applications and issues. *Journal of Basic and Applied Sciences*, *13*, 459-465. 10.6000/1927-5129.2017.13.76.

[14] Zhu, X., & Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, *3*(1), p2-11.

[15] Mishra, V. K., Kumar, S., & Shukla, N. (2017). Image acquisition and techniques to perform image acquisition. *SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology*, *9*(01), 21-24.

[16] Behnke, S. (2003). *Hierarchical neural networks for image interpretation* (Vol. 2766). Springer. p134.

[17] Raj, S., Kumar, S., & Raj, S. (2015). An Improved Histogram Equalization Technique for Image Contrast Enhancement. In International Conference on Communication and Computing. p2.

[18] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, *6*(1), p7.

[19] Gu, S., Pednekar, M., & Slater, R. (2019). Improve image classification using data augmentation and neural networks. *SMU Data Science Review*, *2*(2), p5.

[20] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, *6*(1), 1-48. p8.

[21] Glasbey, C. A., & Horgan, G. W. (1995). *Image analysis for the biological sciences* (Vol. 1). Chichester: Wiley. p1.

[22] Kunaver, M., & Tasic, J. F. (2005, November). Image feature extraction-an overview. In *EUROCON 2005-The International Conference on" Computer as a Tool"* (Vol. 1, pp. 183-186). IEEE.

[23] Gurney, K. (1997). *An introduction to neural networks*. CRC press. pp 13-14.

[24] Zhang, S., Li, X., Zong, M., Zhu, X., & Cheng, D. (2017). Learning k for knn classification. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *8*(3), 1-19.

[25] Srivastava, D & Bhambhu, L. (2010). Data classification using support vector machine. *Journal of theoretical and applied information technology*, *12*(1), p 1-7.

[26] CIFAR-10 and CIFAR-100 datasets. (2021). Retrieved 25 May 2021, from *https://www.cs.toronto.edu/~kriz/cifar.html*.

[27] CIFAR-10 and CIFAR-100 datasets. (2021). Retrieved 25 May 2021, from *https://www.cs.toronto.edu/~kriz/cifar.html*.

[28] STL-10 dataset. (2021). Retrieved 25 May 2021, from *https://ai.stanford.edu/~acoates/stl10/*

[29] The Street View House Numbers (SVHN) Dataset. (2021). Retrieved 25 May 2021, from *http://ufldl.stanford.edu/housenumbers/*

[30] The MNIST database of handwritten digits. (2021). Retrieved 25 May 2021, from *http://yann.lecun.com/exdb/mnist/*

[31] Team, K. (2021). Keras documentation: Fashion MNIST dataset, an alternative to MNIST. Retrieved 25 May 2021, from *https://keras.io/api/datasets/fashion_mnist/*.

[32] ImageNet Dataset. (2021). (2021). Retrieved 25 May 2021, from *https://www.image-net.org/about.php*

[33] Dataset Overview – Cityscapes Dataset. (2021). Retrieved 25 May 2021, from *https://www.cityscapes-dataset.com/dataset-overview/*

[34] Foret, P., Kleiner, A., Mobahi, H., & Neyshabur, B. (2020). Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*.

[35] Tanveer, M. S., Khan, M. U. K., & Kyung, C. M. (2021, January). Fine-tuning darts for image classification. In *2020 25th International Conference on Pattern Recognition (ICPR)* (pp. 4789-4796). IEEE.

[36] Hugo T., Matthieu C., Alexandre S., Gabriel S., Hervé J. (2021). Going deeper with image transformers. *arXiv preprint arXiv:2103.17239*.

[37] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

[38] Harris, E., Marcu, A., Painter, M., Niranjan, M., Prügel-Bennett, A., & Hare, J. (2020). Fmix: Enhancing mixed sample data augmentation. *arXiv preprint arXiv:2002.12047*.

[39] Kabir, H. M., Abdar, M., Jalali, S. M. J., Khosravi, A., Atiya, A. F., Nahavandi, S., & Srinivasan, D. (2020). Spinalnet: Deep neural network with gradual input. *arXiv preprint arXiv:2007.03347*.

[40] Mingxing T & Quoc V. (2021). Efficientnetv2: Smaller models and faster training. *arXiv preprint arXiv:2104.00298*.

[41] Gattal, A., Djeddi, C., Chibani, Y., & Siddiqi, I. (2016, April). Isolated handwritten digit recognition using oBIFs and background features. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)* (pp. 305-310). IEEE.

[42] Gattal, A., Siddiqi, I., Djeddi, C & Al-ma'adeed, S. (2018, August). Writer identification on historical documents using oriented basic image features. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)* (pp. 369-373). IEEE.

[43] Gattal, A., Djeddi, C., Chibani, Y., & Siddiqi, I. (2016, April). Isolated handwritten digit recognition using oBIFs and background features. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)* (pp. 305-310). IEEE.

[44] Browne, M., Ghidary, S. S., & Mayer, N. M. (2008). Convolutional neural networks for image processing with applications in mobile robotics. In *Speech, Audio, Image and Biomedical Signal Processing using Neural Networks* (pp. 327-349). Springer, Berlin, Heidelberg.

[45] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). *Learning internal representations by error propagation*. California Univ San Diego La Jolla Inst for Cognitive Science.

[46] Bouvrie, J. (2006). Notes on convolutional neural networks.

[47] Leibe, B. (2014). Understanding Convolutional Neural Networks.

[48] Hijazi, S., Kumar, R., & Rowen, C. (2015). Using convolutional neural networks for image recognition. *Cadence Design Systems Inc.: San Jose, CA, USA*, 1-12.

[49] Bouchaib, R & ELHami, A. (2018). Introduction to Matlab. 10.1002/9781119492238.app1.

[50] Bogdanchikov, A., Zhaparov, M & Suliyev, R. (2013). Python to learn programming. *Journal of Physics Conference Series*. 423. 2027-. 10.1088/1742-6596/423/1/012027.

[51] Pessoa, T., Medeiros, R., Nepomuceno, T., Bian, G., Albuquerque, V.H.C. & Filho, P. P. (2018). Performance analysis of google colaboratory as a tool for accelerating deep learning applications. *IEEE Access*, *6*, 61677-61685.

[52] Wang, K., Chen, C., & He, Y. (2020, April). Research on pig face recognition model based on keras convolutional neural network. In *IOP Conference Series: Earth and Environmental Science* (Vol. 474, No. 3, p. 032030). IOP Publishing.

[53] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[54] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, *15*(1), 1929-1958.