Parallel study of CNN transfer learning models for object detection

Project Report submitted in partial fulfillment of the requirement for the degree of Master of Technology

in

Computer Science & Engineering

under the Supervision of

Dr. Vivek kumar sehgal / Dr. Rajinder sandhu

By

Arush Kaushal 192203

Jaypee University of Information Technology

Waknaghat, Solan – 173234, Himachal Pradesh

Certificate

This is to certify that synopsis report entitled "Parallel study of classifiers used in conjunction object detectors", submitted by Arush kaushal in partial fulfillment for the award of degree of Master of Technology in Computer Science & Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been made under my supervision.

This synopsis has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Vivek Sehgal

Date: **Supervisor's Name & Signature**

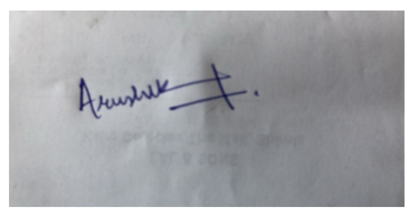
Designation Dr. Vivek Sehgal

Date: **Co-Supervisor's Name & Signature**

Designation

Acknowledgement

I am indebted to Dr. Vivek kumar sehgal and Dr. Rajinder Sandhu for providing this auspicious opportunity to work on this research. I am still deeply grateful for their patience and direction with zeal .The project would not be where it is today without them.. I also thank the Jaypee university of Information and technology for supporting this research.



Date: 8/07/21

Name of the student

Arush Kaushal (192203)

Table of Content

List of Figures

List of Figures

S.No. Title Page No.

Abstract

Computer vision is a branch of computer science that focuses on the recognition and comprehension of images and scenes. Face detection, mask detection are few of examples. In this project, we're using MobileNet,Inception, RetinaNet, and SSD and YOLO, which are all highly accurate object detection algorithms and methods. The area object in an outlined rectangular box can be used to detect any object in an image, define every object, and also designates all tag to it using dependencies. COVID-19 pandemic has had a rapid impact on our daily lives, affecting global commerce and movement. Protecting one's face with a mask has become the new standard. As a result, detecting face masks has become a critical role in aiding global society. TensorFlow, Keras, OpenCV, and Scikit-Learn are some of the basic Machine Learning packages that can be used to achieve this goal. Here proposed method identifies the face in the picture before determining if it is obscured by a mask.

CHAPTER 1

INTRODUCTION

1.1 Introduction

Object detection is common but difficult problem in computer vision. It's essentially a machine vision technology that allows humans to retrieve all of the relevant data with relative ease. People can now easily track numerous objects with high levels of precision and accuracy when it comes to photographs. Advanced picture awareness can aid systems in being practical in a wide variety of settlements from the army to education. So, in general, progress in this area would benefit fields where vast volumes of data are stored in the form of images or clips, such as protection, autonomous vehicles, facial recognition, people counting, and self-driving cars. addition, Object Detection includes image processing and video surveillance. In the field of object detection, a lot of algebra, machine learning, image processing, optimisation, probability, and other things are studied.

The year 2020 has thrown mankind a mind-boggling sequence of events, the most life-changing of which is the COVID-19 pandemic, which has shocked the world since the year began. COVID-19 has called for stringent steps to be followed in order to prevent the spread of disease, which has an effect on the health and lives of many people. From the most basic sanitation codes to hospital care, People are doing all they can to ensure their own and society's safety; one of the personal protective equipment is face masks. When people leave their houses, they put on face masks. Authorities ensure that individuals wearing face masks in groups and public areas are strictly enforced. A policy should be built to ensure that citizens obey this basic safety concept. To verify this, a face mask detector device can be used. Face mask detection refers to determining whether or not an individual is wearing a mask. The first step in detecting the presence of a mask on a person's face is to detect the face, which divides the technique into two parts: detecting faces and detecting masks. Face detection is a form of object detection that can be used in a number of settings, including defence, biometrics, and law enforcement. Detector systems have been built and deployed in every country . All of this science, however, needs optimization; a better, more accurate detector, because the planet cannot afford any more corona events.

1.2 Problem Statement

Object recognition, which is the process of recognising real-world objects in videos or photos, such as bicycles, cars, and people. It is one of the most basic and long-standing problems in computer vision. The description and localization of an object in an image are also part of object detection. The former is concerned with determining the image's class, while the latter is concerned with determining the position of the object in the image, which entails drawing a bounding box around the object. When you account for various perspectives, lighting, deformation, and occlusion, what appears to be a straightforward issue becomes increasingly complicated.

Face mask detection includes data that is firstly trained and then test is done on it to view images with mask and without mask. Furthermore accuracy is compared between different models and model with highest accuracy on precised dataset is choosen for further implementations.

1.3 Organisation

To begin, a brief yet comprehensive analysis of past practises and foundational works that laid the groundwork for modern frameworks has been presented, which covers object classification and object detection techniques prior to the advent of deep neural networks, such as the KNN classifier, SVM classifier, Softmax classifier, Viola Jones, and other machine learning . This research starts with a broad overview of the object detection problem, then delves into three stages: data collection, feature extraction, and object detection. Following that, the article delves into the difficulties that object detection models face.

1.4 Conclusion

This chapter provides a high-level overview of the project. It went through the basic problem of object detection and how it will be dealt with in this article. It also provides a short overview of how the research was conducted. Furthermore, techniques for detecting face masks and their principles of operation.

CHAPTER 2

LITERATURE SURVEY

Object detection is one of the hottest subjects in image processing and visionary computer right now. It is used in a wide variety of industries, from small scale personal applications to large scale industrial applications. An picture is read and one or more objects in the image are classified in object detection. A boundary called the bounding box also defines the position of such objects. Researchers have traditionally used pattern recognition to predict faces based on previous face models. The Viola Jones detector was a revolutionary face detection device that was an optimised technique of using Haar [1], digital image features used in object recognition. It struggled, however, because it did not work well on faces in dark areas or those that were not frontal. Since then, researchers have been eager to design new deep learning algorithms to enhance the models. Deep learning helps one to learn features in an precised way, eliminating need of shape function extractors using prior knowledge. Object detection methods that are dependent on deep learning are classified into two categories: one stage object detectors and two stage object detectors.

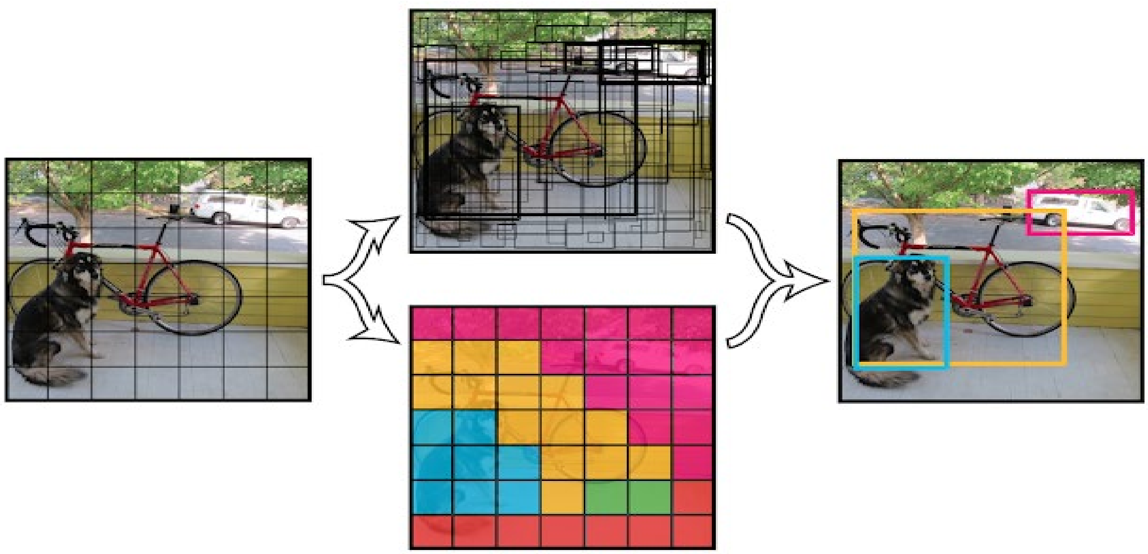


Fig1: Object Detection

Face de-tection, like object de-tection, uses similar archi-tectures as one stage and two-stage de-tectors, but more face features introduced for increasing face detection accuracy. Their is some study that focuses on detecting face masks. OpenCV, Pytorch Lightning, MobileNet, RetinaNet, and Support Vector Machines were used to model some existing facemask detectors. Two ventures will be discussed here. The Real WorldMasked Face Dataset (RMFD), which includes 5,000 faces with mask of 525 people &90,000 normalfaces, was used in one project [8]. These images are 250 x 250 pixels in size, and they are unbalanced in terms of race and ethnicity.

2.1 Object Detection

Object recognition, which cities to identifying physical world objects such as car, vehicles, images of people in videos or photographs, is one of the most basic and long-standing problems in computer vision. The description and localization of an object in an image are also part of object detection. The former is concerned with determining the image's class, while the latter is concerned with determining the position of the image in an object, and drawing a box with boundary around the object. When you account for various perspectives, lighting, deformation, and occlusion, what appears to be a straightforward issue becomes increasingly complicated. But, before we get into different models, we need in considering issues and the challenges it poses. Three categories in which object detection model is divided : informative area collection, feature extraction, and classification.

Informative Region Selection : A variety of approaches have been addressed , ranging from the very crude divide and conquer approach , to exhaustive approaches such as scanning the entire image with a multi-scale sliding window, to advanced deep learning techniques for feature selection.

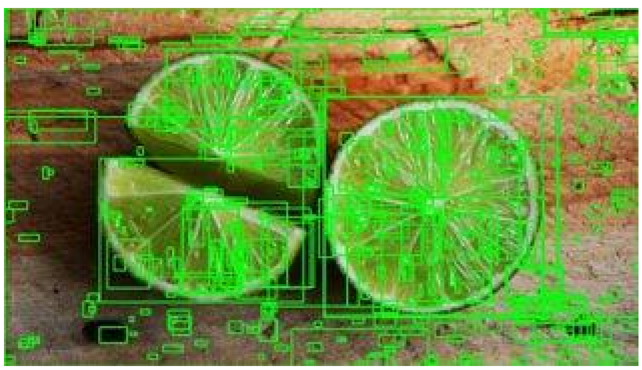


Fig2 : Informative Region Selection

Features Extractions: These algorithms are efficient in their classification capabilities in order to classify a large number of different objects; in other words, certain attributes that distinguish one object from another must be linked or mapped to that object. Every image classification and object recognition model requires feature extraction.

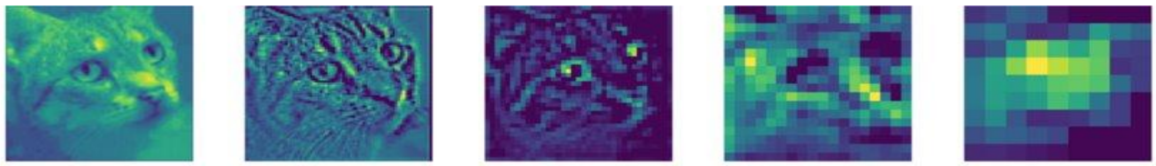


Fig 3 : Extracting features of cat.

Classification Determine which of the various categories are present in the information region you've selected.Some classifiers as SVM very user-friendly and can be easy run on our PCs in a short amount of time. During the last two decades, ample of experimentation been completed on this precised topic, and image classification algorithms did significant gains, as illustrated by Chum and Zisserman . Some of the most widely used robust classifiers are AdaBoost , Deformable Part-Based Model , and Support Vector Machine . In this article, we'll go through some of the most common ones, including KNN-classifiers, SVM classifiers, and Softmax classifiers.

Fig 4: Classification of objects

2.2 Classification

Major difficulties on this problem are precisely explained in session. Image classification is the process of assigning a label to an input image from a set of categories that may or may not be mutually exclusive. This is a long-standing fundamental problem in computer vision that, despite its simplicity, has remained unsolved . It is at the heart of a wide range of practical applications. Object recognition and image segmentation are two other Computer Vision issues.

2.2.1 K-Nearest Neighbor

Most basic image classification algorithm is the kNN algorithm. The images and labels of the training images are simply stored when training a kNN classifier. A similarity index is used to classify a new image, calculated by function such as Manhattan distance or Euclidean distance.

Manhatten Distance L1	$d_1(I_1, I_2) = \sum_p \lvert I_1^p - I_2^p \rvert$
Euclidean Distance L2	$d_2(I_1, I_2) = \sqrt{\sum_p (I_1^p - I_2^p)^2}$

Table 1: index metrices (Similarities)

The Manhatten gap compares the images pixel by pixel and sums up all of the variations.

In Euclidean distance, difference is squared pixel wise,addition of values pixel wise for attaining square root, which provides us the required distances.

The image can be assigned the mark that is dominant among its k-neighbors once the k closest images to the given image using one of the L1/L2 distances have been identified. It becomes more immune to outliers as a result of this.

The Euclidean distance simply squares the pixel wise difference, then sums those values for each pixel to get the square root, which gives us the required L2 distance.

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=15)
knn.fit(x_train,y_train)
y_pred=knn.predict(x_test)
```

2.2.2 Support Vector Machine

It is a linear classifier that uses the decision planes principle to classify date points onto non-overlapping classes. Planes specify decision boundaries which divide data points into mutually exclusive groups. However, it is possible that it will not result in perfect classification; in this case, the SVM classifier will find a hyperplane that maximises the margin while reducing the number of misclassifications A linear classifier establishes a linear relationship between image and label ratings.

$$f(x_i, W, b) = W.x_i + b = W.x_i + b$$

The above equation converts raw image pixels into K mark ratings,yi is the dot. The bias vector b sizing of K x 1, weight matrix W is of size [K x D]. The final result is a matrix sizing of K x 1 which contains class wise scores.

Each rows in W weighted vectors represents prototype for particular groups. As a result, these are essentially similar instructions, assuming that the templates are pre-trained. Our score function is parametrized by this weight vector, which can be modified to match the expected scores in the preparation with the ground reality labels.

In other words, the SVM loss expects the exact class score to be greater then the incorrect class values by a certain amount. If there is a score for any class,

It will lose if it is unable to overcome this margin; otherwise, the loss will be 0 . The SVM loss function's objective is to aid in of weights that satisfy those constraints while also minimising the total loss caused by:

$$L = \frac{1}{N} \sum_{I} L_i$$

```python
from sklearn.svm import SVC
svm = SVC(kernel="linear", C=0.025,random_state=101)
svm.fit(x_train, y_train)
y_pred=svm.predict(x_test)
```

2.2.3 Softmax Classifier

Softmax classifier, which uses a different loss function, is another common classifier. The Softmax classifier is a multiple-class generalisation of a binary logistic classifier. When it comes to interpreting the scores, the key There is a distinction between the Softmax and SVM classifiers. The softmax classifier calculates probabilities for each class that can be easily translated into confidence scores, while the SVM classifier scores have little or no insight into which class score should be prioritised.

We compute loss using the cross-entropy loss and view the scores as unnormalized log probabilities for each class:

y = Li + log efj j + log efj j + log efj j + log

where fj is the j-th class's score .. The aim of this:

P (y | x; W) = f (z) = e Zij (y / x; W) = f (z) = e zj P (y / x; W)

The softmax function is I I j f j e j It squashes the values between zero and one that amount to one from a vector of arbitrary real scores.

2.2.4 Logistic Regression

Logistic regression is described as follows: It is for this algorithm’s model the probabilities which describe the potential output of a every trial.

Advantages: It was created for classification, it's especially necessary for figuring out how many independent variables interact with a outcome variable.

Disadvantages: Only projects when predicted variable is binary, all predictors are unrestrained of one another, and the results have no missing values.

```
from sklearn.linear_model import LogisticRegression
lr =  LogisticRegression()
lr.fit(x_train, y_train)
y_pred=lr.predict(x_test)
```

2.2.5 Naïve Bayes

Based on Bayes' theorem, the Naive Bayes algorithm assumes that every pair of features is independent. Naive Bayes classifiers are useful in many real-world situations, such as document classification and spam filtering.

Advantages: To approx the necessary values, only small amount of training data is required. When compared to more sophisticated methods, Naive Bayes classifiers are extremely fast.

Disadvantages: Naive Bayes is considered as a poor figurer.

```
from sklearn.naive_bayes import GaussianNB
nb =  GaussianNB()
nb.fit(x_train, y_train)
y_pred=nb.predict(x_test)
```

2.2.6 Stochastic Gradient Descent

It is effective method for fitting linear models. It's especially useful ample of samples are present. For grouping.

Advantages: Efficiency and ease of execution are two advantages.

Disadvantages: Requires ample of parameters and is vulnerable for function scaling.

```
from sklearn.linear_model import SGDClassifier
sgd =  SGDClassifier(loss='modified_huber', shuffle=True,random_state=101)
sgd.fit(x_train, y_train)
y_pred=sgd.predict(x_test)
```

2.2.7 Decision Tree

It is generates a set of instructions that can be used for classifying data when a set of attributes and classes is present.

Advantages: It's easy to comprehend , needs minimal planning, and can adjust with both numerical and categorical data.

Disadvantages: It can produce trees that are difficult to normalise, and they can be unreliable since minor changes in the this can create some diffeent type of tree.

```
from sklearn.tree import DecisionTreeClassifier
dtree = DecisionTreeClassifier(max_depth=10, random_state=101,
                               max_features = None, min_samples_leaf = 15)
dtree.fit(x_train, y_train)
y_pred=dtree.predict(x_test)
```

2.2.8 Random Forest

Ensemble Method: When we take multiple machine learning algorithm to create one big machine learning algorithm.

It basically combines lot of Decision tree method and runs them many times which provide us with it.

Advantages: minimal amount of over fiting and, in most situations, Decision tree accuracy is very less as compared to Random Forest.

Disadvantages: Implementation is tough and difficult algorithm are all disadvantages.

```python
from sklearn.ensemble import RandomForestClassifier
rfm = RandomForestClassifier(n_estimators=70, oob_score=True, n_jobs=-1,
                             random_state=101, max_features = None, min_samples_leaf = 30)
rfm.fit(x_train, y_train)
y_pred=rfm.predict(x_test)
```

2.3 Factors affecting Object Detection

When we consider the factors discussed below, the seemingly basic issues of image classification . Detecting objects in images with sufficient lighting conditions in a limited database of objects, on the other hand, is relatively simple and considered solved by the computer vision community. The developed models are challenged by occlusion and variable illumination.

2.3.1 Position

Objects in a picture may be placed differently than they were in the training data, and the model must account for this.

2.3.2 Scale

Regardless of how the object's size changes due to different views,all objects byobject detection model should be detected precisely and efficiently.

2.3.3 Illumination

Several illumination conditions, that can occur at variable times of the day and under variable climatic conditions, can render entities in picture unrecognisable. If your model relies on color-based recognition, this can be an issue.

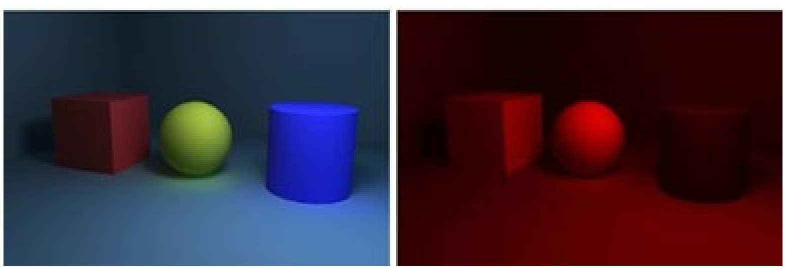


Fig 5: Illumination

2.3.4 Rotation

The model must detect different orientations of the same object since they belong to the same object. The object's orientation must not obstruct identification of the object in the picture.

2.3.5 Occlusion

Simply stated, a portion of an object can be obscured by another obstructive object or the object itself, making detection and classification difficult. According to [12], occlusion can be divided into three types: inter-object occlusion, self-occlusion, and context scene structure occlusion.

Fig 6: Occclusion

2.4 History of Deep Neural Network

Since it attempts to imitate the human mind, cognition is a prerequisite for any design built on artificial intelligence's core premises. Cognitive computing is based on a human brain model that aims to recognise a real-world concrete object, as well as its scope and purpose. On a digital image or video, there are two methods for performing object detection. Deep learning and machine learning are two types of artificial intelligence.

Item detectors based on machine learning were handcrafted or designed prior to breakthroughs in field of DNN. This method incorporates in the algorithm the object's distinct and distinguishing characteristics.

The Viola-Jones Algorithm was the first machine learning-based object detection algorithm, and it is the basis of the OpenCV library. In 2001, Paul Viola and Michael Jones came up with the algorithm. Edge-features, line features, and four rectangles are included in the Viola-Jones [4] algorithm. The algorithm reduces the amount of data to be processed and speeds up the algorithm by converting the original image into a grayscale image. It then moves a sub-window in very small steps from the top-left corner to the bottom-right corner, looking for haar-like features.

The algorithm's performance is improved further by cascading.In sub windows cascade detects efficient features. it transfers the sub-window without looking for weak features, saving computation power while increasing detection rate.

The Histogram of Gradient is a common machine learning technique. It is an entity descriptor that works remarkably well in both stable image and video to human detection. This algorithm extracts features from every part of the image.

By gathering information about the gradient, an object in an image can be detected. We divide the image into multiple cells and create a histogram gradient for each pixel within the cells. When we combine these histograms, we get a descriptor. On the local histograms, contrast-normalization is used.

The value of normalisation is determined by the image block's strength. SVM [11] is used to detect the object class using these features.Object detection can be performed using deep learning-based approaches without the need for complex features to be defined. Machine learning-based approaches are easily outperformed by methods in this approach. CNN is commonly used in deep learning methods.

The visual cortex of the human brain inspired convolutional neural networks. When humans see an object, different neurons fire, each one responsible for detecting different features such as lines and edges.Convolution, max pooling, and a completely linked layer can be used to emulate this visual cortex behaviour.

2.5 Artificial Neural Network

A device that copies the person brain and nerve system is known as an ANN. No of layers make up an artificial neural network. The input layer is the input giving layer in this, and from output layer we can obtain the required results. There occurs a layer between these both layers that layer is called as intermediate layers.

Hidden Layer are those which are not visible. These layers are made up of nodes termed neurons, these neurons are linked to others through synapse.

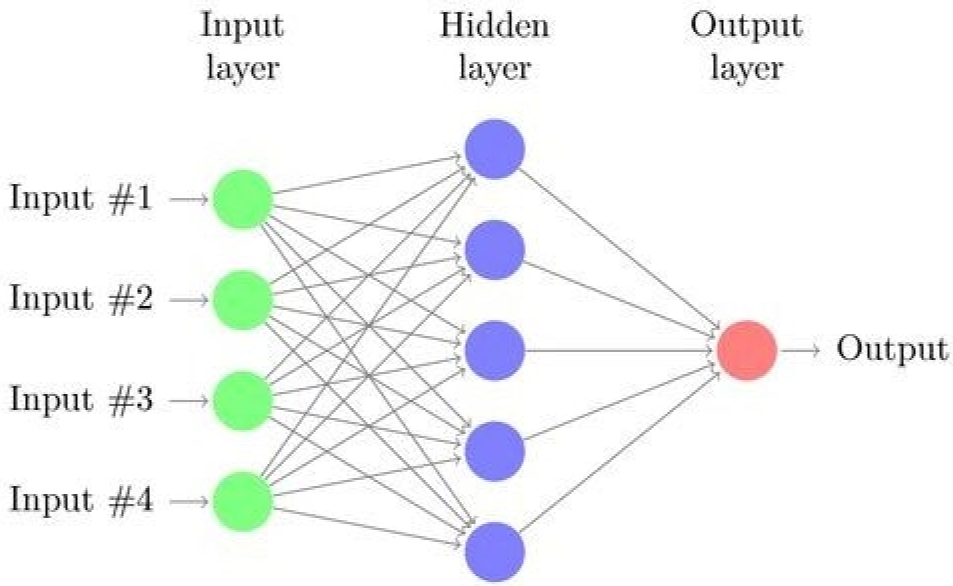


Fig 7: Architectural design of ANN

2.5.1 Cost Function

The neuron's expected values are compared to the real value using a cost function. In training the model, the cost function is extremely important. If the real value of the neurons is ‘y,' and the expected value is ‘a,' then

$$Z = w \,.\, x + b$$

$$\sigma(Z) = a$$

Activation function is represented by letter ‘'.

The quadratic cost function is:

$$C = (\Sigma\,(y - a)^2) / n$$

When the cost function's value is low, our model is more accurate since the variance between the expected and actual value is smaller. Only ‘w' can be modified out of the parameters ‘w', ‘x', and ‘b', as ‘b' is the bias for any neuron .

The weight of the neuron(synapse) 'w' should be balanced in such a way that the cost function is minimised to make our model more effective and predict an accurate outcome. The approximate value of the cost function is fed into the neural network, allowing the neurons to adjust their weights.

2.5.2 Gradient Descent

It is an optimizing algorithm that is for reduce a function's cost. In essence, it determines whether the cost function's slope is positive or negative by checking its slope at.

If a non-convex cost function is subjected to gradient descent, the local minimum will be chosen. , stochastic gradient descent is used to resolve this error, that rationalize weights after each value is processed, as opposed to it, which rationalize values after all input values goes through this process.

2.5.3 Back – Propagation

Back-Propagation is a technique for rapidly adjusting weights across a network. After the data has been processed, we measure each neuron's error contribution. Back track through network from its path to check and update errors, it heavily relies on the chain law.

The network is fed the training data many times because it is difficult to train the model by running through the data only once. An epoch is a unit of time used by our network to process training data. We should still redo more epochs to make our model understandable.

2.6 Convolutional Neural Network

Traditional neural systems aren't ideal for image processing, so images with lower resolution should be encouraged.

Learning and classification are the two components of a convolution neural network. Convolution, ReLU, and Pool are used in feature learning, while flatten, completely connected plate, and softmax are used in classification.

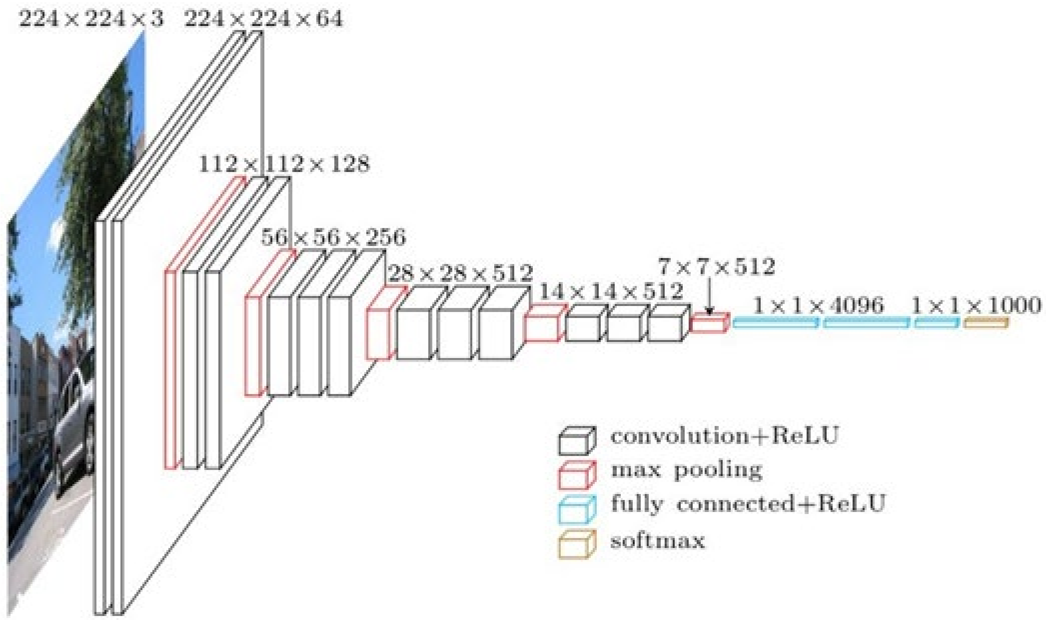


Fig 8: Convolutional Neural Network Architecture

2.6.1 Convolution

The aim of a convolution layer is to extract features from a source image. Convolution maintains the relationship between pixels by learning image features using small squares of input data. Convolution reduces the size of the input image, which is important because it reduces the number of input neurons in the input layer. The mathematical function convolution takes image matrices and kernels as data.

The filter matrix is passed with a fixed stride over the input image, multiplying the corresponding values of the filter matrix and input image.

An image matrix of dimension: (H*W*D)

A filter: FH* FW * D

volume dimensional results: (H − F+ 1) * (W− FW* 1).1

Stride is number of shifts over input metrics. We pass filter one pixel at a time while we make a stride of one. If a filter doesn't match the object, we got two choices for that as padding of 0s to make the image fit, or remove that part where filter is not fitting. Other choice is known as true padding since it holds a validated portion of the image, while the last one is known as zero padding.

2.6.2 ReLU- (Nonlineearity)

Rectified Linear Unit, which is defined as:

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

ReLU's motivation is to introduce non-linearity. Because of its superior efficiency, ReLU [13] is favoured over other non-linear functionalities.

2.6.3 Pooling Layer

Pooling layer is used to minimize number of parameters. Spatial-pooling is a technique for reducing the size of a function map while retaining important information.

There are three methods for spatial-pooling:

Max-Pooling: The highest value from function map is used. Average Pooling: The function map's average value is used.

In the pooling layer, a max pooling filter of size 2x2 with a stride of 2 is commonly used. We have the following for the pooling layer:

Enter the following information:

W 1 * H1 * D 1 = W 1 * H 1 * D 1 = W 1 * H 1

W 2 * H 2 * D 2 would then be the output.

H2 = W2 = W2 = H2 = W2 = H2 = W

D2 = D1 (W 1 F)/S + 1 (H1 F)/S + 1

2.6.4 Fully Connected Layer

In fully connected layer we are basically adding whole artificial neural network to convolutional network. Moreover in CNN, hidden layers are fully connected layer.

Basically fully connected layers are last few layers, which compile data extracted from previous layers to form final output.

2.6.5 Softmax

It is an activation functions used in completely connected layers last layer to classify the input image's generated features into different groups.

2.6.6 Convolutional neural network and object detection

Since the no. of events of the objects isn't fixed, a traditional CNN can't be used as an object detectors. We can solve this problem by selecting a various part from images, using a CNN for detecting artefacts inside segment itself. Key issues in approach is items in the image may have different spatial areas and aspect ratios.

2.7 Region based Convolution Neural Network(R-CNN)

Using R-CNN, object detection is a difficulty that can be resolved. Applying selective search, 2000 regions called area proposed are extracted from that part. After being wrapped into a rectangle, these two thousand area proposals are used in CNN, which outputs a 4096 dimension features vectors. Convolutional neural network acts as a extractor of characteristics, and that are stored in the dense output layer. After that, the digged features are fedded into a Support Vector Machine, which classifies the object's existence within the candidate area proposal.

2.7.1 Problems with R-CNN

As nil training takes place during the selective searching algo, that can result in poor candidate area proposals. In addition, because of the large number of area proposals, training the network takes a long time. It can't be applied in real time, either.

2.7.2 Fast R-CNN

It solves speed problem of R-CNN by sharing its convolution layer computation between various schemes and reversing the order in which regions proposals are produced and running the CNN. Photos, R-CNN.

The convolutional function map is used to classify area proposals, which are then wrap in square. The ROI pooling layer resizes shape as they can be fed to the completely connected layer with a fixed scale. Chracteristics in ROI is changed into a tiny HxW function map in the RoI layer, where both H and W are tunable hyper-parameters.

Softmax layer predicts the proposed region's class from the RoI function vector, as well as the bounding box's offset value.

Fast R-CNN outperforms R-CNN in terms of time, since R-CNN feeds the CNN with two thousand area proposals. The convolution operations are peered in Quick R-CNN .

2.7.3 Faster R-CNN

To generate area proposals, Quick R-CNN [15] uses a selective search algorithm, that consumes huge amount of time and a lot of computational resources.

Instead of using selective quest, a RPN is used to generate region selection in Faster R CNN. When compared to Fast R CNN [15], Region Proposals are less computationally intensive, and they share the most computation with the Object Detection Network. The anchors are produced and ranked by the Region Proposal Network, and only the anchors with a high probability of containing an object are sent to the detection of objects network. Area proposal is used by it is to detect and predict object's class. Anchors are cemented in place.

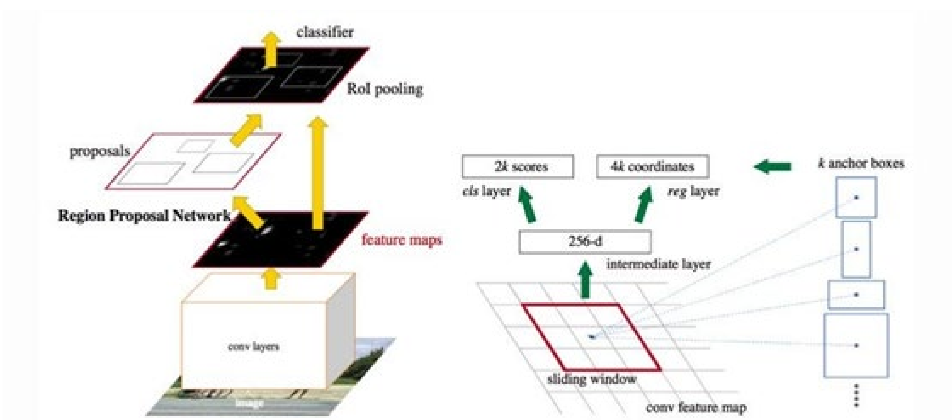


Fig 9: Faster R CNN models illustration.

RPN generates a large number of proposals, which are inspected by the regressor and classifier to determine the existence and occurrence of artefacts. It determines whether the anchors are in the foreground or background.

2.7.4 Pooling ROI

Basically that’s difficult to create a neural system which may effectively dealt at feature of various sizes, but the Region Proposal Network provides convolutional feature maps of various sizes. With the aid of Region of Interest Pooling, convolutional feature maps are reduced to the same scale.

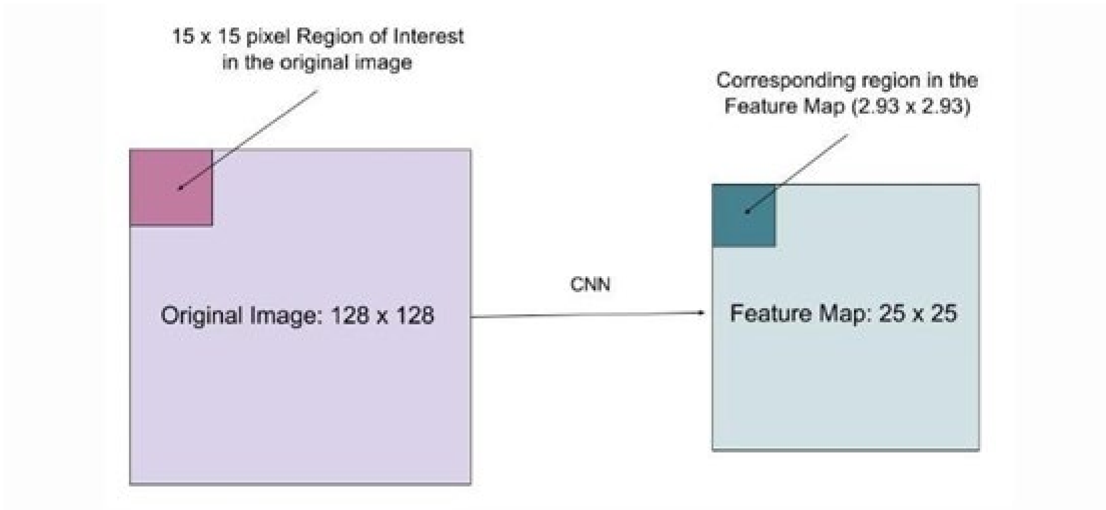


Fig 10: Without rounding off, a ROI mapping done from 1st integer to function map

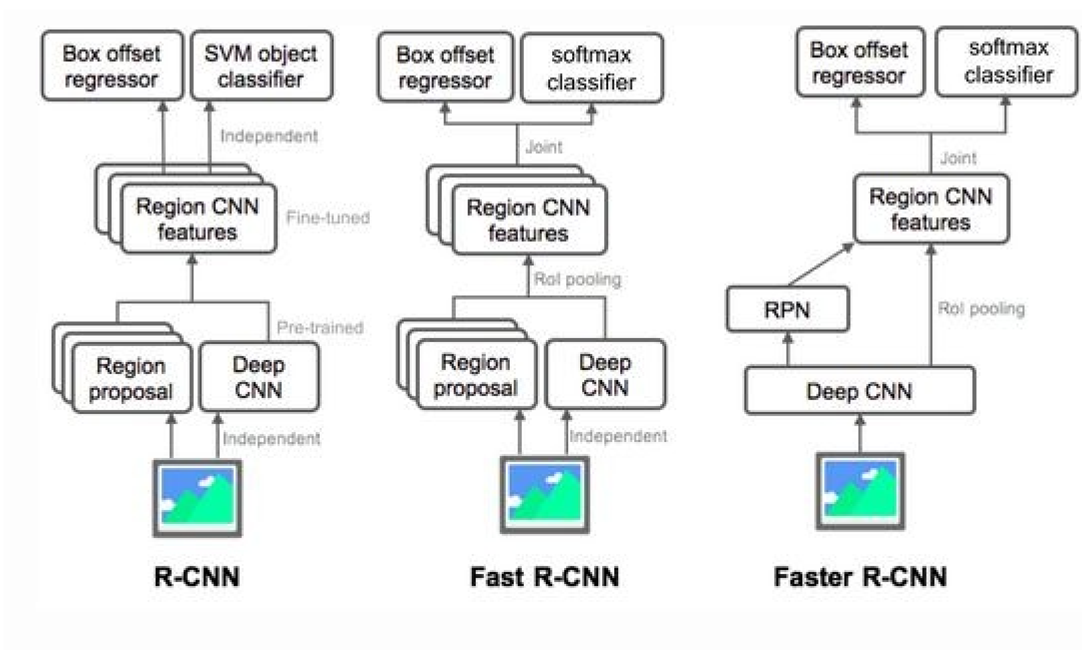


Fig 11: Models summary in R CNN

2.8 Single Shot MultiBox Detector

It identifies and classifies objects in an image using a single network. Unlike the Faster R-CNN [3] object detector, which anticipates the existence of an object using Region Proposal, the Single Shot MultiBox Detector uses a bounding box and modifies it.

The network's final two layers, which are responsible for predictions for increasingly smaller bounding boxes, perform a variety of bounding box predictions. The Ultimate prediction is the sum of these predictions.

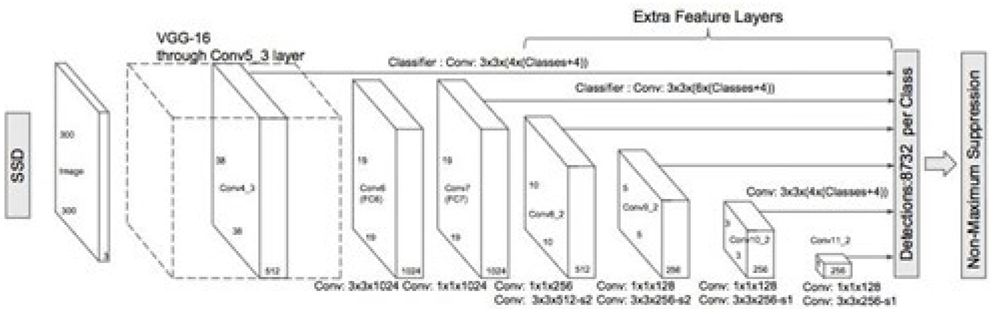


Fig 12: Single Shot MultiBox Detector Architecture

2.8.1 MultiBox

Szegedy et al. developed the MultiBox technique for bounding box regression in Single Shot MultiBox Detector[2]. A bounding box coordinate is suggested by MultiBox. MultiBox comes with two essential components:

Geographical Displacement (l) This value refers to the difference b/w predicted and ground truth bounding boxes.

The following formula is used to measure Multibox box Loss(m), which shows how accurate our overall prediction is:

$$m = c + \alpha l$$

where alpha is used to counteract the effect of position loss.

MultiBox tries to get closer to the ground reality, but detection starts from the beginning.

2.9 YOLO (You Only Look Once)

It is a detector of objects that employs the novel approach for solving object detection difficulties. Unlike traditional objects detectors, YOLO detects objects using classifiers. Each NN used to determine boxes and probability of classes in one pass, saving a significant amount of computing time and energy. Unlike Faster R-CNN, which can be difficult to optimise, a single neural network allows for more optimization.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Fig 13: Darknet-53

The function extractor is the latest Darknet-53 architecture. It's mostly made up of skip-connected [3 by3] and [1 by1] filters. Despite the fact that the formulated architecture attains the results of classification as the original, it is 2x as easy.

[13 by13], [26 by26], and [52 by52], YOLO [1] makes estimates on three different scales. Furthermore, it determine six bounding boxes as size of grid. Multiple detections of the same object are suppressed using NMS suppression.

Furthermore, due to its simple device design, it is extremely easy to implement, giving it an advantage over its competitors. However, compared to other detectors such as Faster-RCNN [3], There are far too many localization errors in this fast object detector. YOLO [1] is a good general-purpose object detector because of the smooth tradeoff between speed and accuracy, as well as the potential for optimization.

2.10 Conclusion

It addressed few of pioneering chores that settled the groundwork for frameworks that are currently trending, as well as some approaches that existed before the advent of DNN, . It also provided a quick rundown of today's object detectors, as well as a description of how they work.

CHAPTER-3

SYSTEM DEVELOPMENT

It focuses heavily on method of implementing the algo, criteria to compare various algorithms, medium on which it was implemented, and the languages used to write them.

3.1 Specifications: Software and Hardware

Its needed to develop, to run the algorithm are listed below.

3.1.1 Specifications:Software

Algorithms, both in Jupyter notebooks (*.ipnyb files) and *.py files, were written in Python 3.6. due to Python's strong interpreter, through which we can debug fastly and portion of code to be executed partially, this was a deliberate option. Not only that, but the limited syntax prevents the reader from being engrossed in the syntactic specifics of a language.

Python 3.6 enables quick execution in Jupiter notebooks, eliminating requirement to recompile the programme for each minor shift.

Furthermore, the deep learning system PyTorch was written in Python, as its name suggests. PyTorch isn't just another library written in another language; it's designed to work hand-in-hand with Python code. The following are some of PyTorch's advantages over other deep learning frameworks:

1. PyTorch makes the most of CPUs and can also use graphical processing units to speed up computations.

2. PyTorch is highly optimised and has a low computational overhead.

3. PyTorch's support for complex neural networks gives it an advantage over the competition. This is critical when we need to programmatically adjust the behaviour of our network.

4. Machine learning and deep learning algorithms are simple to learn and use.

5. It is open source.

Some libraries that were included:

1. numpy: Its-array processing kit that's widely used.

2. matplotlib: matplotlib is a plotting library that produces high-quality diagrams, charts, and figures in a number of formats.

3. torchvision: In the field of computer vision, this python package offers a user interface for obtaining and working with common datasets, model architectures, and image transformations.

These packages must be mounted on the computer that will be used to run or evaluate the algorithms. The programmes may not result in the desired behaviour if these packages are missing. They are completely necessary for the modules to function as planned and defined here.

These packages are already installed on Google Collaborate, and the code is ready to use.

3.1.2 Specifications: Hardware

The implemented algorithms are platform agnostic, meaning they can run on any Windows/Ubuntu/Mac platform. However, the computing capacity might not be sufficient to meet requirements. In order to avoid your device crashing due to inadequate memory requirements, the given should be executed on Google collaboratory.

Per gmail user gets access to a private cloud with 12GB of RAM and a Tesla K80 GPU for computing. All local info, on the other hand, is wiped out every 12 hours.

3.2 Datasets

The dataset we used contains a total of 3835 images, of which 1917 are masked faces and 1920 are without mask faces. Images were collected from the Bing SearchAPI, Kaggle dataset, and RMFD dataset. Portion of photographs from all three sources is the same. The photographs depict a wide range of races. The ratio of with mask to without mask faces determines balance of datasets. This dataset must be partioned into three portions: training, test, and validation . prevention of overfitting is main aim of split. The term "generalisation" refers to a model's ability to perform well on a dataset it has never seen before (test data). We use a subset of the entire dataset to train the model, which we call the training set. This data is observed and learned by the model, which then optimises its parameters. To selecthy perparameters, the validation dataset is used (learning rate, regularisation parameters). It may stop understanding with its training data till model outperforms on our validation data. It is a subset of data that is used to provide an favourable evaluation of a last model that sets on the training data. The splitting ratio, that is highly dependent on type of models we're creating and the data itself, is used to split data. If our dataset and model necessitate a significant amount of testing, we use a greater portion of the data solely for training, as in our case.

3.2.1 Dataset Preprocessing (CIFAR-10)

This dataset contains 60,000 32by32 pixel colour files. There are 6000 images in each of the ten object groups. A total of 50K training images and 10K test images are available.

Splitting training set in two parts—a validating set and a testing set—is best way to tune hyperparameters of k-NN, SVM, and Softmax classifiers. When compared to the training set, the validation set is significantly smaller. The 50,000-image training set is divided into a training set of 49K images and avalidation set of 49K images for purpose. This set is basically a dummy test set for larger-parameter tuning.

Fig 14: Dataset of CIFAR-10

3.2.2 Dataset Preprocessing(COCO/PASCAL VOC)

The bounding boxes for several datasets follow a consistent framework. Assuming autonomous driving, the data will be divided into two clutters: for automobile bounding boxes and another for pedestrian bounding boxes. The former will, of course, be strongly centred in the image's middle.

Fig 15: Dataset of PASCAL VOC

It leads in enhanced object detection accuracies; fro accomplishing this, clustering is used to classify the top k-boundary boxes using k-mean clustering. Anchors are the term for these boundary-boxes. The YOLO detector is the only one who uses anchors.

3.3 Train Object detector

It implies that our training data must contain the exact response, so that the output that is expected could get tested against the exact output to strengthen the model when it predicts wrong predictions. steps below can be used to train an object detector in general.

1. Set the weights of the neurons to a value similar to zero at random.

2. The dataset's observations are then fed into each input layer.

3. According to their weights, subsequent neurons are triggered by the influence of preceding neurons.

4. The values are propagated until they enter the output layer, where they are converted to the network'y' prediction value. The discrepancy between the expected and real value is then computed.

5. Backpropagation employs the chain rule to measure each neuron's contribution to the generated error and adjusts the weights accordingly to minimise it.

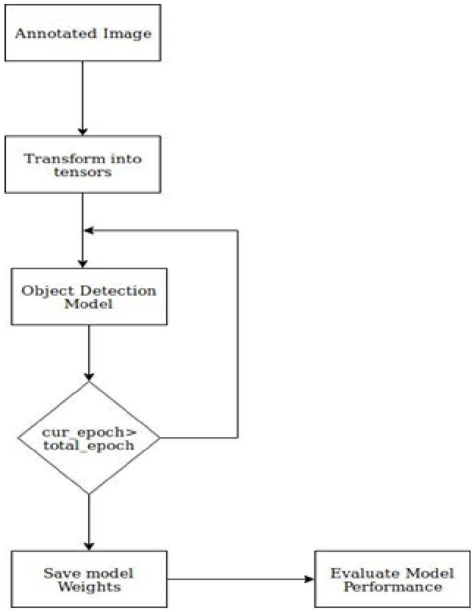


Fig16: Object Detection Pipeline

3.4 Architecture

The Single Shot Detector algorithm works with an input picture that has a bounding box set against the elements. The well-known convolution model is used in the process of prediction of an object in image. A collection of revert boundingboxes with various shape and size, aspected ratios estimated for every pixel in image. Moreover, a confidence value for every possible entity are calculated for all pixels, with additional remark as "No Object." This calculation is carried out for a variety of function charts. To obtain feature maps, we typically use pre-trained techniques that are commonly used for high-quality classification problems. this part of this research is termed as base model. Our base models for the SSD are the VGG-16 network. Comparison is done between these both ground truth value and training.

VGG-16 is a dense network of 16 layers of convolutional layers that can be used to extract features in order to identify and detect artefacts. The architecture4 was chosen because it consists of stacks of 3x3 kernel size convolutions that thoroughly extract various featureinformation, as well as max-pooling and ReLU to transfer details flow in the model and add non linearity respectively from the given picture. It uses 1x1 convolutionblocks for additional nonlinearity without changing the spatial dimension of the input. There are several weight parameters that end up giving an improved output due to the small size filters striding across the image.

The working functionality of SSD is depicted in the block diagram [10]. TheVGG-16 is being used as the base model at the input end. At the end of the base model, several additional function layers are introduced to handle offsets and confidence scores of different bounding boxes. The layers are flattened at the bottom of the figure to allow predictions for various bounding boxes. Finally, non-maximum suppression is used to exclude duplicate or nearly identical bounding boxes around the same objects. There could be times when a neighbouring pixel estimates a bounding box for an object with less confidence, and the object is ultimately rejected.

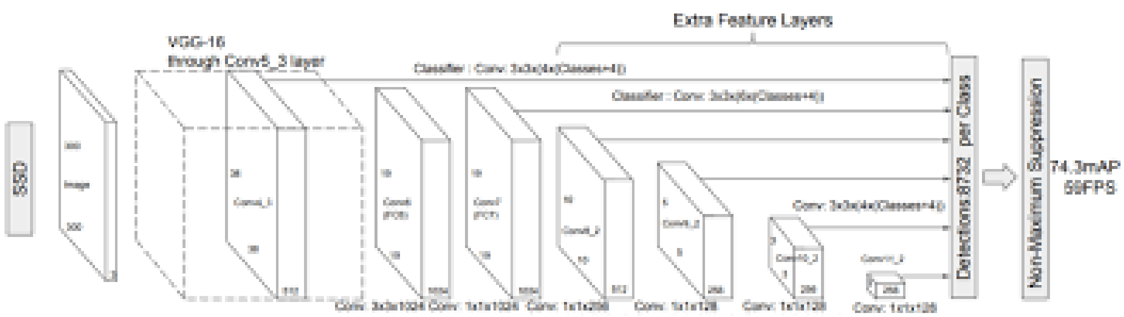


Fig 17: Working of SSD

The problem can be broken down into two parts: first, detecting the presence of multiple faces in a each stream, and afterthat, in another, detecting the presence or absence of a masks on the profile. The new version of OpenCV provides a A pre-trained convolutional neural network for face detection is included in the Deep Neural Network (DNN) module (CNN). The new edition increases face recognition accuracy as compared to previous versions. When a new test image is shown.

We train the second part of the model using a dataset that includes images with and without masks. Keras and Tensorflow were used to train our model. For the base model, the first step of training entails storing all picture labels in a Numpy array and reshaping the corresponding images (224, 244, 3). Augmenting images is a very useful technique. because it expands our dataset by adding images from a different angle. We performed the following image augmentations at random before inputting: circular movement upto 20 degree, focusing inside and outside up to 15%, width or altitude change till 20%, and counterclockwise shear angle up to 15 degrees.

Transfer learning, which means using a model that has been pre-trained on millions of labels previously, is now a standard practise for image classification, and it has been shown that this approach results in a significant improvement in accuracy. Clearly, the presumption is that both problems are sufficiently similar. It employs a well-structured and deep neural network that was trained on a massive data set. Since the issue is similar in nature, we can use the same weights that can extract features and later convert those features to artefacts in the deep layers.

MobileNetV2 was used as the base model, assigned to ‘Image_Net' weighs. Image_Net is a massive image database that has been trained on hundreds of thousands of images, making it incredibly useful for image characterization. We truncate the head and use a sequence of self-defined layers for the base model. We used a 50 percent dropout layer for optimization, a dense layer with output shape(None, 128) and activation ReLU, and finally another dense layers.

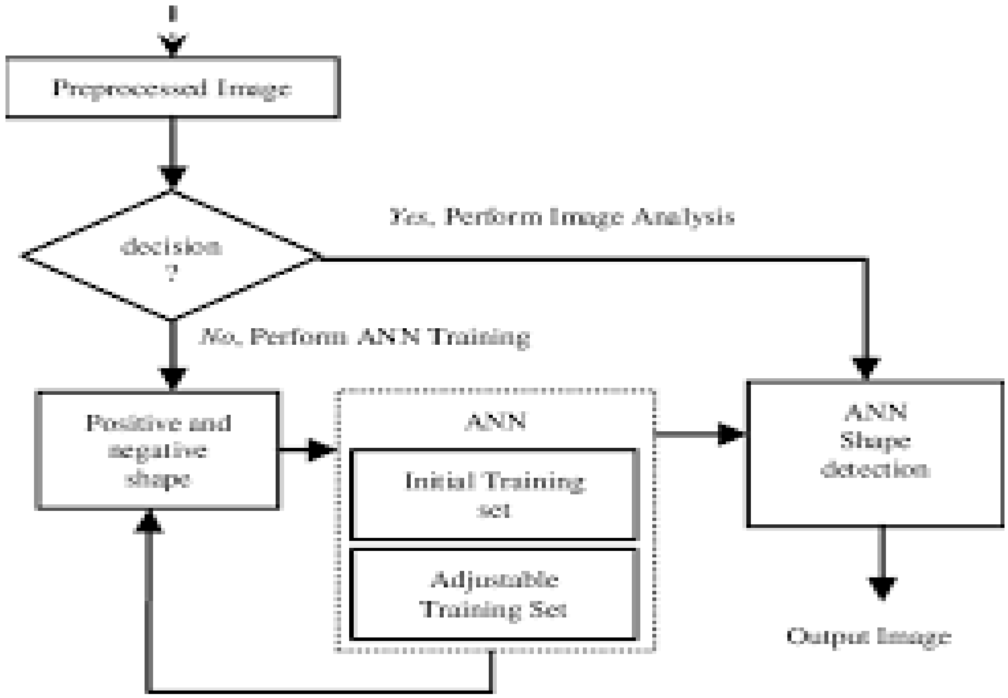


Fig 18: Flow diagram Object Detector

3.5 Training

During preparation, we compare the default bounding box with ground truth boxes of different shape size and ratios, furthermore, use the Intersection over Union (IoU) method to pick the best matching box for each pixel. IoU determines how much of our expected box corresponds to fact. The values range from 0 to 1, and increasing IoU values decide prediction accuracies; the best value is the highest IoU value. IoU's equation and pictorial representation are as follows:

$$B1B2B1B2 = IoU(B1, B2) = B1B2B1B2$$

3.6 Hyper Parameters

A hyperparameter is a parameter or variable that must be set before an algorithm can be applied to a dataset. These parameters define the model's "High Level" properties, such as its complexity and how quickly it can learn. Before the actual training phase starts, hyperparameters are set. There are two types of hyperparameters: optimizer hyperparameters and model hyperparameters.

Before we start the actual training phase, we can tune or optimise our model using optimizer parameters. The following are some popular optimizer hyperparameters. The learning rate is a hyperparameter that controls how often the weights of our neural network are adjusted in relation to the gradient. Epochs are the hyperparameters that decide how much the model is run.

We first practised with different hyperparameter values, changing one while holding the other unchanged, and recording the results in each case. Using measurement metrics, we chose the hyperparameters that resulted in better results. The hyperparameters were chosen as follows: the initial rate to learn fixed to 0.001, the size of batch fixed to 32, and the number of epochs was fixed to 20. In our case, target size is one of the hyperparameters that we held (224, 224, 3) because it is MobileNetV2's default input form.

3.7 Loss Function

The overall detection problem's loss can be divided into two categories: localization loss and confidence loss. The difference between the default expected boundingbox and the G.T. boundingbox is the fixation loss (g). We try to adjust the shape of the box for given centre (cx, cy) in order to eradicate the lose.

Confidence loss is the just measure of how high the probability of the presence of an object is, whenthere exists an object. Similarly, the localization loss is the measure of how much a predicted box differs from the GT box. Our model tries to minimize every loss by predicting presence of object and then correctly classifying it to the right class.

CHAPTER 4

PERFORMANCE ANALYSIS.

4.1 Dataset

Standardized dataset like CIFAR-10, PASCAL-VOC and COCO dataset, and image segmentation (COCO) include CIFAR-10, COCO [17], and PASCAL VOC [18], allowing for testing and comparing precision of various method.

They do not, however, provide any valuable details about the inference time of the various techniques used.

They do not, however, they aint having any valuable details about the various techniques and inference time.

4.1.1 Dataset:(CIFAR-10)

This dataset contains 60,000 coloured images with a 32by32 pixel size. There are 6000 images in each of the ten object groups. There are 50000 images for preparation and 10,000 images for testing. The following are the classes:

Fig 19: Random Samples from CIFAR-10

There are no photos that overlap between cars and trucks in any of the groups since they are mutually exclusive. Big trucks are classified as "Truck," while sedans and SUVs are classified as "Automobile." Pickup trucks are not used in any of the two classes.

4.1.2 Pascal VOC Dataset

There are no pictures in any of the classes that overlap with cars and trucks because they are mutually exclusive. "Truck" refers to large vehicles, while "Automobile" refers to sedans and SUVs. In none of the two groups are pickup trucks used.

Fig 20: Samples of Pascal VOC

4.1.3 COCO Dataset

COCO dataset consists of 2,00,000 images and 80 classes if object classes is included in its train, validation, and test sets. Annotations are present on all object instances in the image, allowing the predictions to be validated. The majority of models use a softmax layer for object classification, assumption is that classes present in it are mutually exclusive. Its labels , on the other hand, are not mutually exclusive. As a result, instead of using a softmax layer, logistic regression is used for its training purpose .

1

Fig21: COCO Dataset

Fig 22 : COCO Dataset

4.2 Classifier:(k-NN, SVM, Softmax)

When these classifiers were evaluated onto test data with 10,000 images after being trained on the CIFAR-10 dataset, the following results were obtained.

Classifier	Accuracy
k-NN	29.79%
SVM	37.29%
Softmax	36.6%

Table 2: Results of CIFAR_10 Classification

The k-NN classifier performs poorly when evaluated on the training set. Furthermore, predicting the mark for any given picture takes O(N) time.

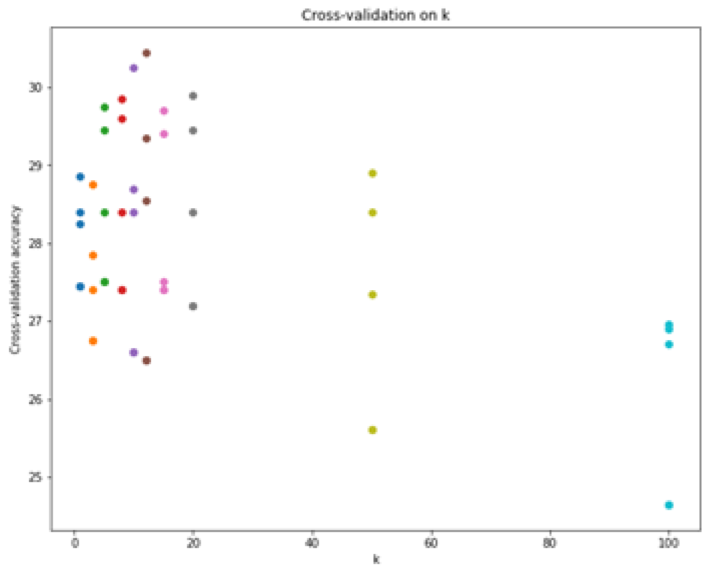


Fig 23: Cross-validation of k-NN on k

The performance of SVM is better than k-NN classifier, only by a small margin. Furthermore, the SVM classifier's expected scores do not indicate how strongly the image belongs to a specific class.

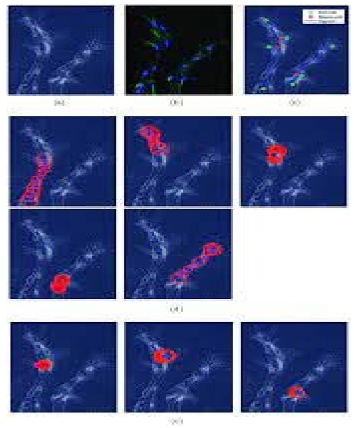


Fig 24: SVM Classifier Templates

The Softmax classifier is a near competitor to the Support Vector Machine, but there is fiction of difference between these both. The benefit for use of softmax over Support Vector Machine is softmax classifier's determined scores are probability distributions for the training dataset's specified groups. However, where the groups are not linearly separable, the softmax classifier cannot be used.

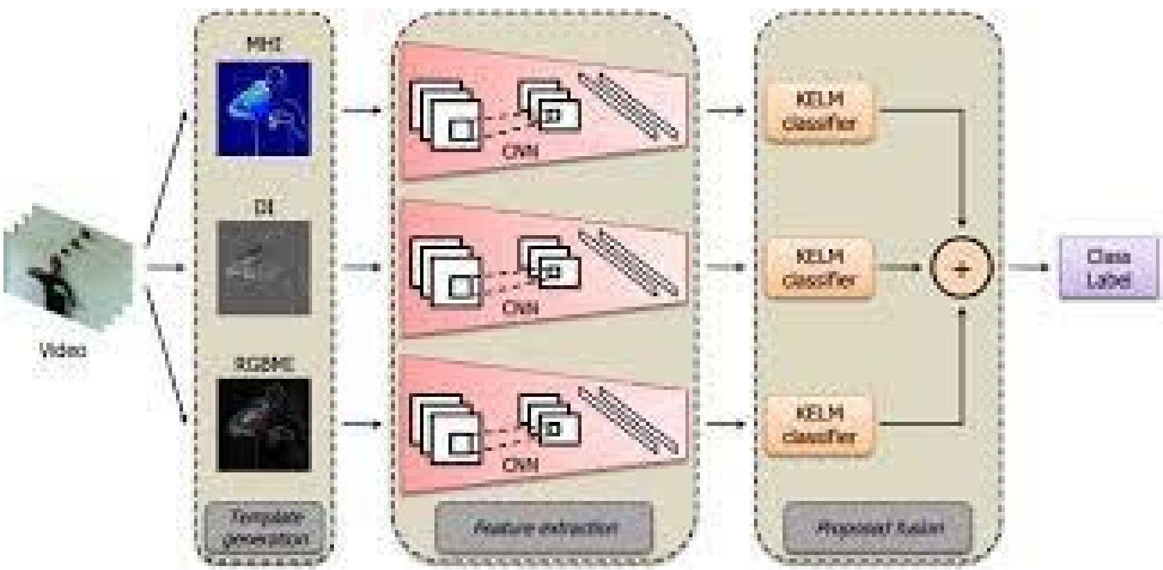


Fig 25: Softmax Classifier Templates

4.3 VGG-16 Classifier

The single shot multibox detector is based on a slightly improved version of VGG16, and it can provide insight into how SSD performs classification on the CIFAR-10 dataset.

The classification results are shown below after training the neural network architecture for 20 epochs with a batch size of 512 on the CIFAR-10 dataset.

The rate of learning is reduced to 0.01 from 0.1 as our accuracy starts to plateau after 10 runs through the dataset.

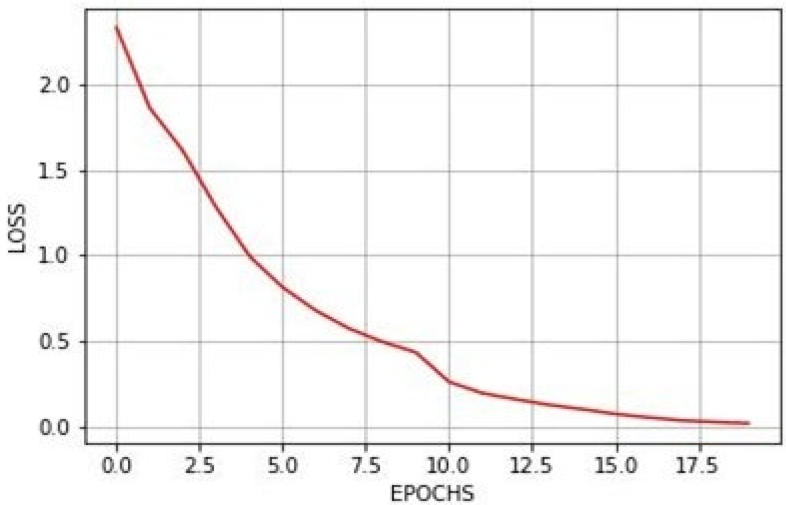


Fig 26: VGG 16 Loss Graph Vs Epochs

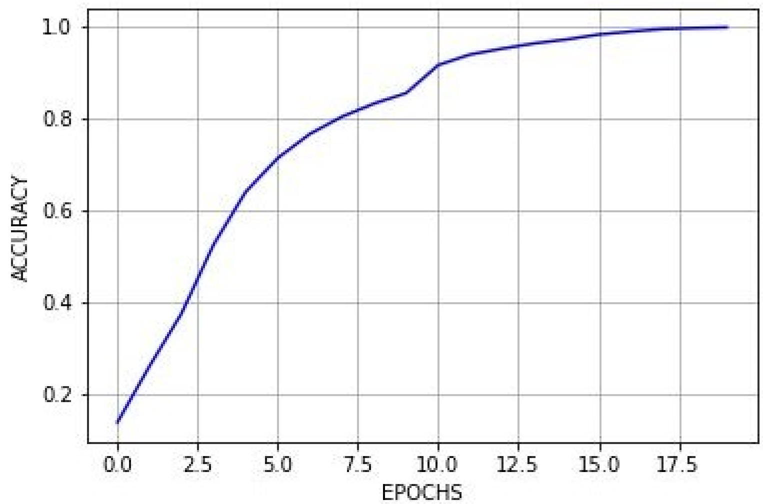


Fig 27: VGG 16 Epochs vs Accuracy

As seen in the graphs above, with the processing of classifier in data, accuracy rises and there is downfall in loss .

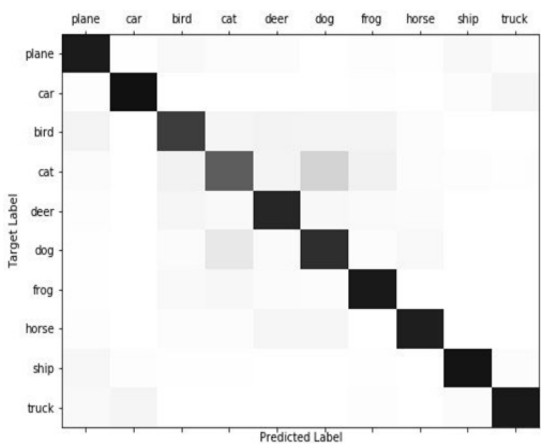


Figure 28: CIFAR-10 (VGG-16 Confusion matrix)

4.4 Classifier(DarkNet)

Backbone of YOLO detector is Darknet, Now checking how it ooutperforms on CIFAR-10 dataset. The classification results of YOLO on this dataset provides us with idea how good YOLO is working.

The results we get after training this N.N. on the CIFAR-10 dataset for 20 epochs. The rate of learning is 0.1, and in the next (10 epochs) =0.01.

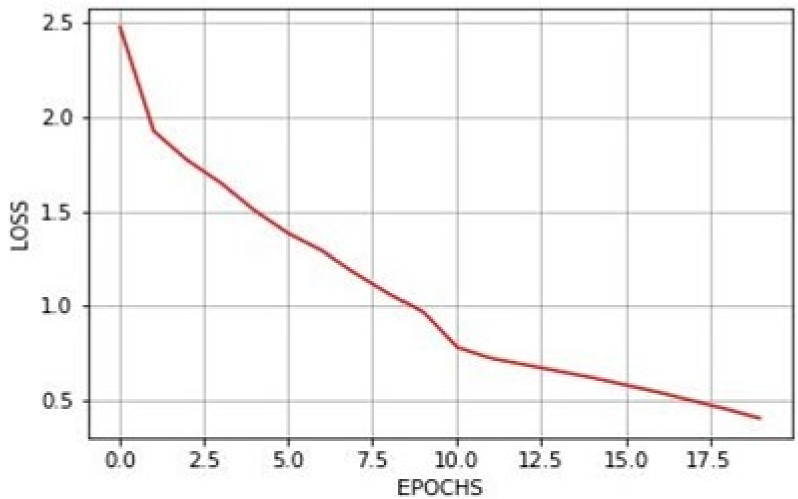


Fig 29: Darknet Classifier(Loss vs Epochs)

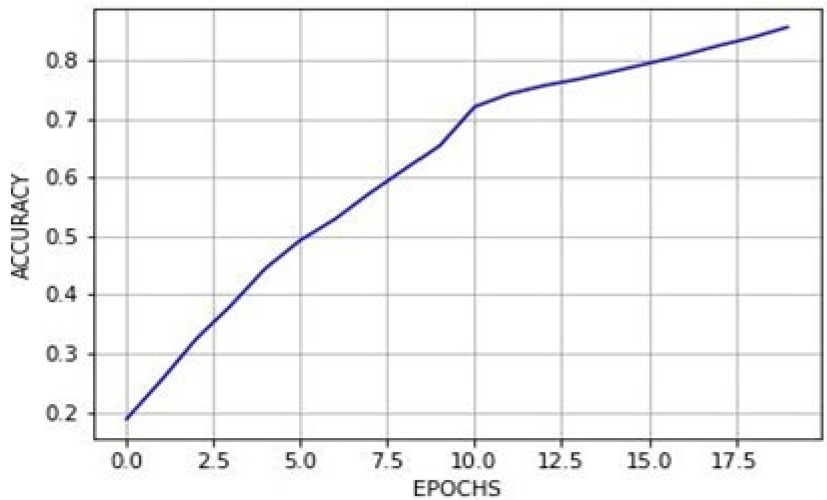


Fig 30: Darknet Classifier(Accuracy vs epochs)

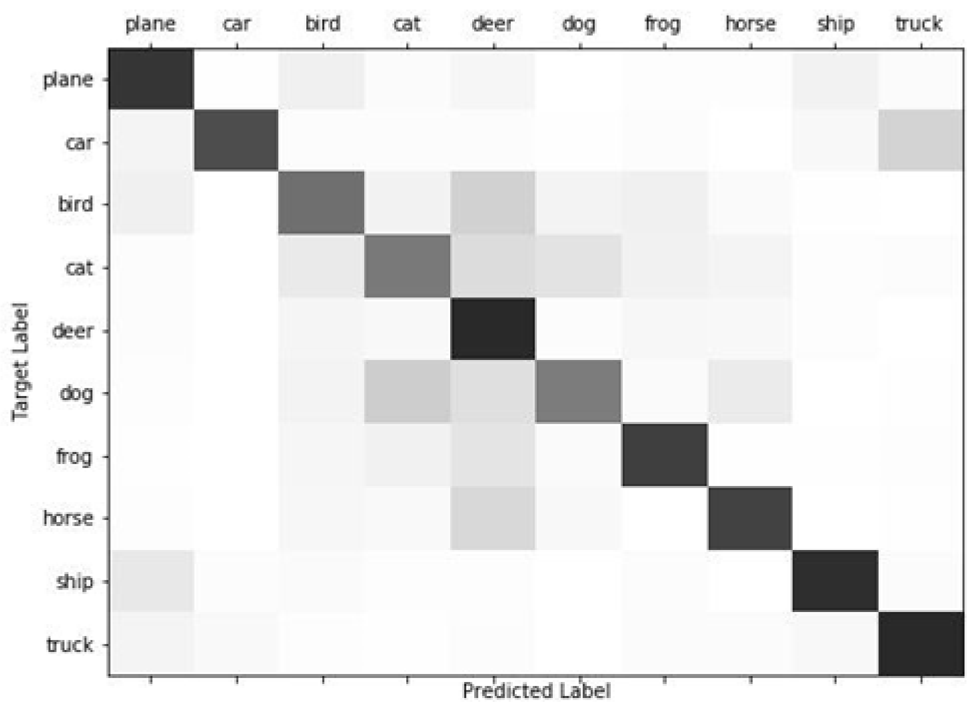


Fig 31: Confusion Matrix on CIFAR-10

The Darknet classifier, like the VGG16, appears to perform well in most classes, with the exception of few classes. It misinterprets lke dog with cat and vice versa, as well as horses and deer, owing to their brown coats. Cars are often often labelled as trucks. Once educated, it easily outperforms (k-NN, SVM, and Softmax) by a broad margin and is competitive with the results provided by the VGG-16 classifier.

Standards	**Epochs**
Maximum	356.72
Minimum	352.63
Average	353.74

Table 2: Epochs(Darknet)

4.5 Study of Classifiers in Comparison

For all classes, the table below compares the accuracy of various classifiers on the CIFAR-10 dataset. As can be seen from the table, classifiers that use CNN for feature extraction outperform those that depend on template matching.

Standards	**VGG16(Epochs)**	**Darknet(Epochs)**
Max Time	28.83	353.72
Min Time	27.53	350.61
Avg Time	28.08	351.77

Table 3: Epochs(VGG-16 and Darknet)

Classes	**DarkNet-19**	**VGG-16**	**k-NN**	**SVM**	**Softmax**

Aeroplane	88.5	88.2	44.8	38.7	46.7
Cars	86.6	91.7	11.2	37.6	28.3
Birds	65.7	72.3	41.3	13.9	8.9
Cats	69.2	62.4	17.2	17.4	5.4
Deers	81.1	84.2	41.1	27.4	20.2
Dogs	61.3	83.2	12.5	28.5	31.3
Frogs	92.2	88.2	24.3	64.2	62.3
Horses	83.2	87.6	16.4	28.1	25.4
Ships	87.7	92.1	72.3	57.8	58.5
Trucks	88.5	88.9	8.4	54.2	61.4
Mean Accuracy	**80.75**	**84.53**	**27.64**	**37.9**	**36.6**

Table 4: classifier's accuracy.

The Darknet-19 classifier takes longer to process the dataset than the VGG16 due to more layers present in it, other one with only 16 layers in comparison to it.

4.6 SSD(Result Analysis)

The used VGG 16 architecture to implement the previously described object detection model, SSD[2]. The following are the results of training :

Fig 32: Occulated Objects using SSD

Output shows, the machine also holds a wide range of lighting conditions, but it also detects artefacts even when they are partially obscured. The two cars have different levels of lighting conditions; one is brightly lit, and other one behind it is occluded and in the dark, but it is still detected.

When large and small objects were present, however, larger objects dominated, as seen in Fig 14. This could explain why the average precision of smaller objects is lower than that of larger objects.

4.7 Comparative analysis of YOLO and SSD

Fig 33: Input Image

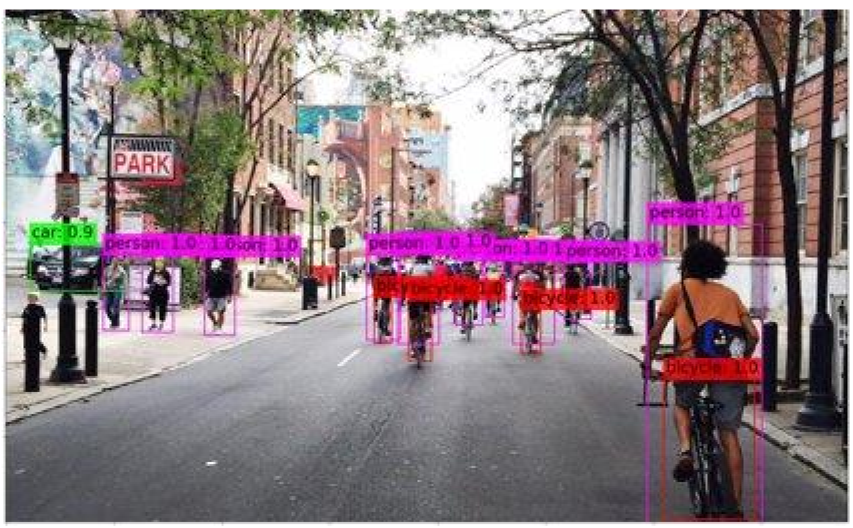


Fig 34: YOLO Output

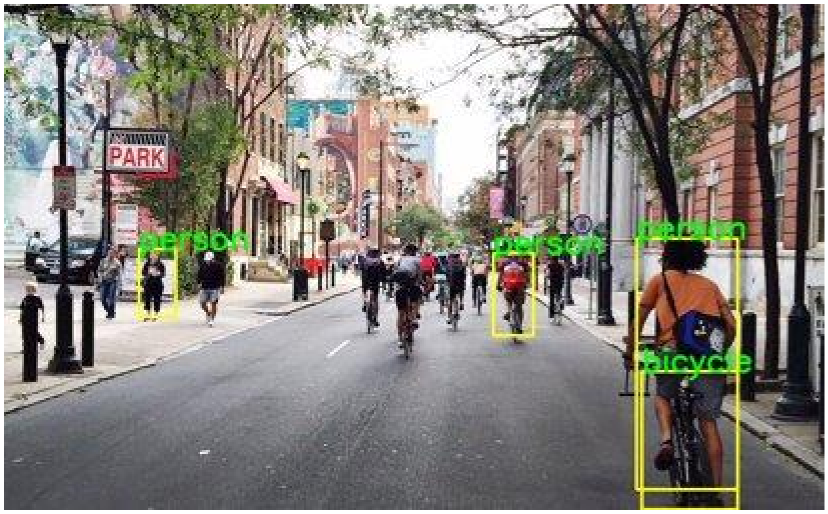


Fig 35: SSD Output

It can be deduced from the above images that YOLO has greater reliability as compared to other as it can detect more occulated objects. Moreover, excellent performance for detecting small objects too.

Standards	SSD Images	YOLO Images
Maximum	2.134	1.89
Minimum	1.421	1.45
Average	1.532	1.712

Table 5 : Inference Timings

From the above findings, it can be deduced that YOLO is more capable of detecting smaller objects in images where larger objects predominate. The above results also show that, while SSD is more versatile when object features are slightly altered, YOLO is more accurate when the image contains a large concentration of objects. It's safer for photos with a lot of occlusion.

CHAPTER-5

EVALUATION

5.1 Testing

For detecting 'mask' or 'no mask,' we tried three different base models. The aim of the exercise was to find the best model for our situation. The assessment process begins with a review of the classification study, which provides information on accuracy, recall, and F1 ranking. These three metrics' equations are as follows:

Precision = True +ve / +ve + False +ve

Recall = True +ve /+ve + False -ve

Accuracy = True +ve +True -ve/(+ve) + (-ve)

We may determine which model is the most efficient using these three metrics. The train loss, validation loss, train accuracy, and validation accuracy are plotted in the second section, which aids in the selection of a final model. The outcomes of various options are shown below.

5.1.1 Xception

"Extreme Inception" is the full form of Xception. The Xception architecture [11] is essentially made up of depth-wise separable convolution layers with residual connections. These convoluted layers are stacked one on top of the other in a straight line. This architecture is simple to create and alter. When compared to Inception V2 and V3, Inception V2 and V3 are much more difficult to define.

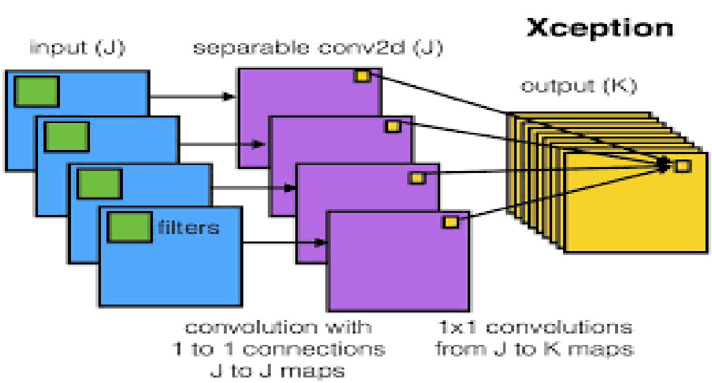


Fig 36: Xception Architecture

5.1.2 MobileNetV2

MobileNetV2 is a bottleneck depth-separable convolution construction of basicblocks with residuals architecture [12]. There are two kinds of blocks in it. The first one is a one-stride residual block. The second is a residual block of stride 2 that is used to downsize.

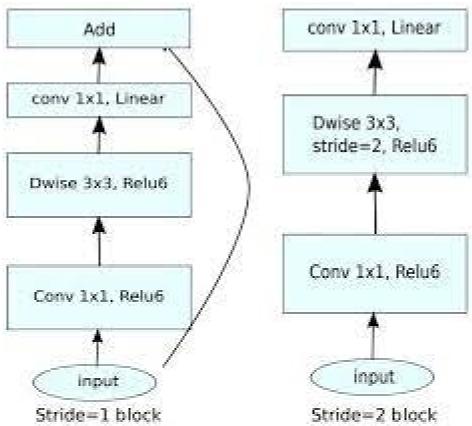


Fig 37: Convolutional Blocks Of MobileNetV2

Both blocks have three layers, as discussed in the layer section. The first is ReLU6's 1x1 convolution. The second layer contains depth-wise convolution, and the third layer contains a1x1 convolution but without any non-linearity.

Network	Top 1	Params	MAdds	CPU
MobileNetV1	70.6	4.2M	575M	113ms
ShuffleNet (1.5)	71.5	**3.4M**	292M	-
ShuffleNet (x2)	73.7	5.4M	524M	-
NasNet-A	74.0	5.3M	564M	183ms
MobileNetV2	**72.0**	**3.4M**	**300M**	**75ms**
MobileNetV2 (1.4)	**74.7**	6.9M	585M	**143ms**

Fig 38: Classification report of MobileNetv2

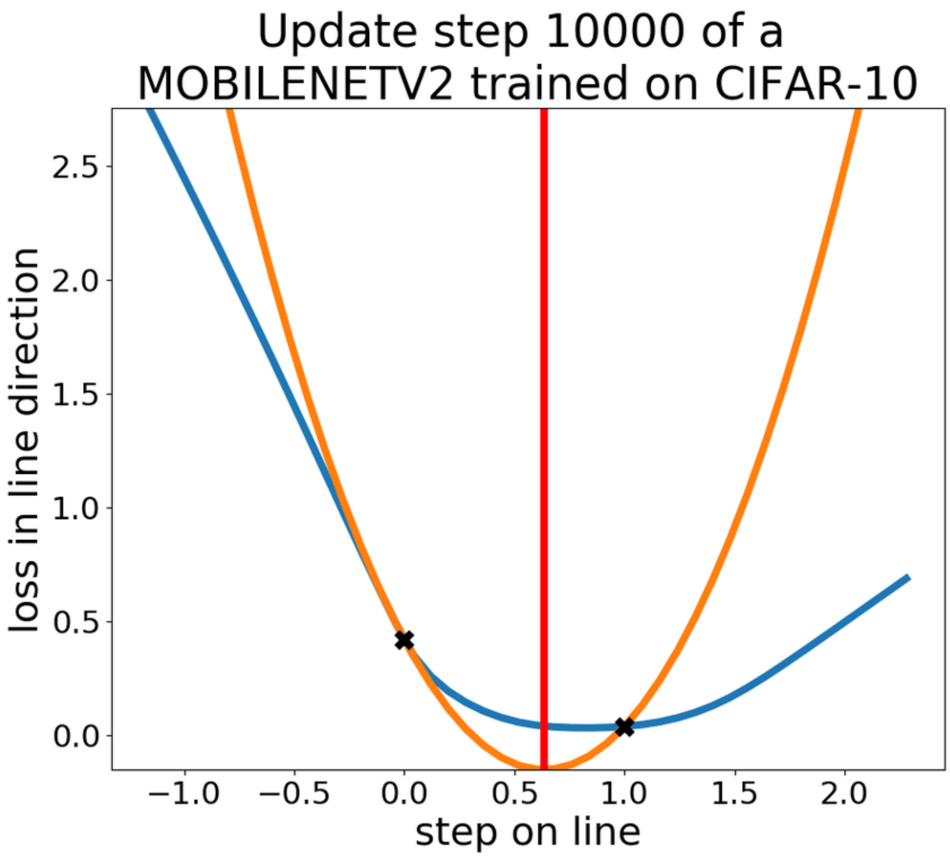


Fig 39: Loss graph for MobileNetV2

5.1.3 ResNet50

ResNet50 is a network of 50 residential communities. It's a three-layer deep convolutional neural network. This three-layer block is implemented by the bottleneck class. We can train millions of images from the ImageNet database and then load them as a pre-trained version of a network. This network will categorise images into a variety of object categories, including face, car, bike, and a variety of animals.

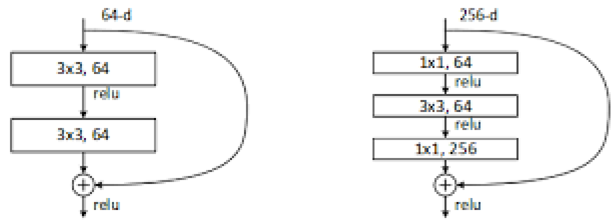


Fig 40: Implementation architecture of Resnet 50

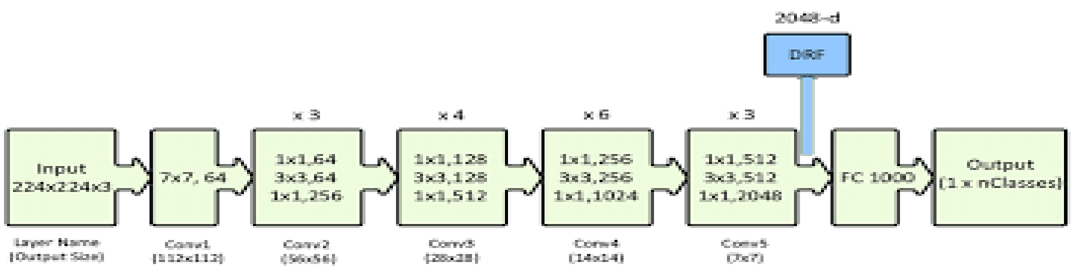


Fig 41: Resnet50 architecture

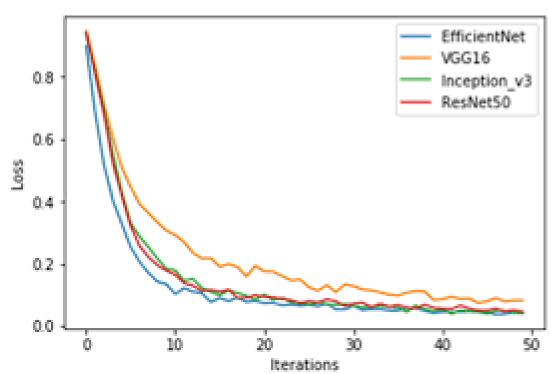


Fig 42: Loss Function in Resnet 50

After looking at these figures, we can deduce that ResNet50 had the worst performance of the bunch. Xception and MobileNetV2 both performed similarly, and the best of them was MobileNetV2's 100 percent accuracy of unmasked faces. Precision is the main metric to determine between these models because our detector can take no risk by incorrectly assuming that an unmasked face is a masked face. We chose to use MobileNetV2 as the base model for our face mask detector algorithm because it provided the best results.

5.2 Inference

Our algorithm was tested on images of one or more faces. By removing and wearing masks one by one, we even introduced it on videos and live video streams. The following are some screenshots of the results:

Fig 43: Person not wearing mask and person wearing mask.

5.3 Training Details

SRCNet had two primary stages of preparation: To train network of SR and condition of wearing mask recognition network trained.

The aim of this Net is to re-establish facial information, that can be accomplished using CelebA package. Preprocessing of images were completed to mimic quality of images in dataset based on the Medical Masks Dataset's characteristics. CelebA first filtered the high-resolution processed images with a Gaussian filter.

Initialization was the prepration of face mask wearing conditions . ImageNet dataset is used to train the network,training values proposed in next move was to create a model for facial identification. The class have been changed to correspond to the numerical figures (10,562). The final updated completely connected layer's weight and bias were initialised. Every epoch, the training data set was shuffled, till 50 epochs training was performed.

Facemask-wearing condition recognition network was fine-tuned using transfer learning, with the final completely linked layers and classifiers changed to fit the classes. In comparison to other initializers, the weights and biases in the last layer were separately sampled from a distribution of zero mean, which yielded superior performance. The optimizer was chosen as Adam, and threshold for rate of learning was set to 104. A 104 weight decay for L2 regularisation is used to prevent overfit problem. The network training for a total of 8 epochs with a batch size of 16 and a total of 16 epochs.

Data augmentation can help to minimise overfitting and improve the final accuracy of the model.

Range for revolution of train set was fixed to 10.(in a regular organisation), moved in area of various sensors like 2020, 21, 5236 12 of 23and 8 pixels, and horizontally flipped for every to train the general facial recognition network. The augmentation was mild during the fine-tuning stage, with rotation within 6 (in normal distribution), shifting by up to 4 pixels (vertically and horizontally), and a random horizontal flip in each epoch . The network was trained for a total of 8 epochs with a batch size of 16 and a total of 16 epochs.

5.3.1 Comparision of Face Mask Wearing Conditions in a Network

Contrast with some CNNs, such as Inception_v3 , Dense_Net , and MobileNet-v2 as the with mask identification, are used for demonstrating benefits and few reasons.

For image processing time, ResNet50 , DarkNet19 , Xception , and VGG19 were tested. The efficiency of a network generally improves as the width of the network increases. MobileNet_V2 demonstrated excellent real-time identification efficiency while requiring minimal storage space and running time.

Furthermore, MobileNet-v2 had a lot of depth, which led to its final success in detecting facemask wear. In our tests, MobileNet-v2 did not perform worse than compared networks, a last inference precision difference of 1% in comparison to the different networks . All of the assessements were carried out by matlab 2020, an i-7 CPU.

DISCUSSION

Our research provided a novel algorithm for detecting facemask wearers that included four key steps: Face detection and resizing, SR, and facemask detection are all examples of image pre-processing.

Identifying the issue: CNNs can achieve higher accuracy by using SR before classification, according to the findings. Furthermore, our evaluation demonstrated that these approaches might be useful to classify these situations, with possible applications in COVID-19 disease prevention. The Medical Masks Dataset and large-scale facial image datasets were used in this analysis.

A new NN was put forward for SR network, which included enhancements to the various purposes and the depth of the connections. In comparison to previous approaches, these advances resulted in significant changes in field of increase in performance efficiency and accuracy, as measured by SSM. The last result of this network was viewed using variable resolutions, it was stored in network and more useful informations were added to it for incresing the performance. It basically enhances its efficiency by integerating with face mask recognition. SRCNet used techniques of preprocessing to improve efficiency by removing unnecessary values such as backdrop, variable contrasts. Furthermore, pre-processed images may be used to achieve better facial recognition. During situation when face is covered with mask recognition n/w preparation, understanding was also used. Finally, in 3 subsets, SRCNet obtained a 97.90 percent precision and performs standard methods of classification using last images, the SR network by more than 15% in kappae. They were also demonstrated in an ablation experiment. The robustness of SRCNet was demonstrated by displaying similarity results in various kinds of masks with various colors. Moreover, approachs like k-NN and support vector machine were contrasted and evaluated, with the SR network outperforming them, according to our findings. The detection of facemask-wearing conditions resembles facial recognition in several ways.

The creation of a facemask-wearing condition recognition network, on the other hand, is difficult for a variety of reasons. One of the most significant challenges is the lack of datasets. In comparison to general facial recognition datasets, these are typically limited and picture resolution is poor. Furthermore, the challenge of identifying people who wear facemasks wrongly increases significantly. To address these issues, SRCNet was created, which combines an this netwpork to classification. SR network resolved the difficulty of less qua, and the challenge of using a small dataset of numerous wearing-facemask-incorrectly examples was solved by transfer learning; both methods significantly improved efficiency. There haven't been any deep learning research on facemask-wearing condition recognition to our knowledge. Mask constraint was detected with 98.65 percent precision in our analysis, suggesting that SRCNet has a lot of potential to help with automated facemask-wearing condition detection. SRCNet is built on effective Convolutional Neural Networks for this condition recognition, which takes into account network complexity. SRCNet's has low computation requirements for people dealing with (IoT) technology, it's useful for encouraging people to wear facemasks correctly for disease prevention.

FUTURE WORK

Work in the Future Face masks have recently become mandatory in more than fifty countries around the world. In public places such as malls, public transportation, offices, and shops, people must cover their faces. Technology is sometimes used by retailers to count the number of customers who visit their shops. They would also like to monitor impressions on digital displays and advertising screens. Our Face Mask Detection tool will be improved and released as an open-source project. Our programme can be used to identify people without a mask using any current USB, IP or CCTV cameras.

CONCLUSION

For facial image classification, a new facemask-wearing condition recognition system was put forward, which merges networks especially SRCNet. The input images were preceeded with image pre_processing, image classification , SR, and mask wearing requirements to know weather mask is put on or face is without mask, recognition to identify facemask-wearing condition. Finally, SRCNet outperformed conventional in kappa last image identification method is used, achieving a 98.70 percent accuracy.Our findings tells that the preceeded SRCNet can recognize facemask-wearing conditions by precision, which is essential in public security over this pandemic situation.

Our research has a some constraints. To begin with as some dataset sensors were used which showed results that to prove wheather mask is used the results to predict it was tough as it don't considers all constraints like environmental factors etc. Moreover, to predict it on video work becomes more cumbersome that cant be checked precisely. The detection time for each image is not that fast as required like only 20 images in 2 second,which creates difficulties in case of videos. A more comprehensive collection, includes photographs , that gonna be obtained and analyzed in future studies.

Faces in various postures, settings, and lighting conditions must be included in the data collection. This network will be made more efficient, a better robust and reliable a technique be investigated, that may help with realistic implementation of detecting this constraint. It will also show better efficiency if images that are present in dataset are trained in accordance to fulfill the requiremnents, which may help with the realistic implementations of detecting mask condition.

The Softmax classifier is a near competitor to the Support vector machine classifier, there leaves a very little gap between their performances. The benefit of using a soft-max classifier over Support Vector Machine is that the softmax classifier's predicts scores are probability distributions for the training dataset's specified groups. However, where the groups are not linearly separable, the softmax classifier cannot be used.

When larger objects dominate an image, we can infer that YOLO is much efficient and reliable when we need to detect small objects. The above findings also show that, while Single shot detector is more robust when features are changed, YOLO is much accurate, whenthe image contains a large number of objects.

REFERENCES:

[1] Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, 2015.

[2]S. Zhai, D. Shang, S. Wang and S. Dong, "DF-SSD: An Improved SSD Object Detection Algorithm Based on DenseNet and Feature Fusion,IEEE Access, 2020

[3]J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. arXiv preprint arXiv:1506.02640, 2015.

[4]Romain Vial, Hongyuan Zhu, Yonghong Tian and Shijian Lu “Search Video Action Proposal with Recurrent and Static YOLO”,2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, ISBN (e): 978-1- 5090-2175-8, September-2017.

[5]M. Buric, M. Pobar and Ivasic-Kos, “Object Detection in Sports Videos”, 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics(MIPRO), Opatija, Croatia, ISBN (e): 978-953-233-095-3, May-2018.

[6]Z. Akata, F. Perronnin, Z. Harchaoui and C. Schmid, "Good Practice in Large-Scale Learning for Image Classification," in IEEE Transactions on Pattern Analysis and Machine Intelligence,2018

[7]Lele Xie, Tasweer Ahmad, Lianwen Jin , Yuliang Liu, and Sheng Zhang ,“ A New CNN-Based Method for MultiDirectional Car License Plate Detection”, IEEE Transactions on Intelligent Transportation Systems, ISSN (e): 1524-9050, Vol-19, Issue-02, Year-2018.

[8]Haihui Xie, Quingxiang Wu and Binshu Chen, “Vehicle Detection in Open Parks Using a Convolutional Neural Network”, 2015 6th International Conference on Intelligent Systems Design and Engineering Applications(ISDEA), Guiyang, China, ISSN (e): 978-1- 4673-9393-5, August-2015

[9]Syed Mazhar Abbas and Dr. Shailendra Narayan Singh, “Region based Object Detection and Classification using Faster R-CNN”, 2018 4th International Conference on Computational Intelligence and Communication Technology(CICT), Ghaziabad, India, ISBN (e): 978-1- 5386-0886-9, February-2017.

[10]Yin-Lon Lin, Yu-Min Chiang and Hsiang-Chen Hsu, “Capacitor Detection in PCB Using YOLO Algorithm”, 2018 International Journal on System Science and Engineering(ICSSE), New Taipei City, Taiwan, ISBN (e): 978-1-5386-6285-4, December-2017

[11]Wang, Y., Pan, Z., & Pan, Y. : A Training Data Set Cleaning Method by Classification Ability Ranking for the k-Nearest Neighbor Classifier. IEEE Transactions on Neural Networks and Learning System 2019

[12]L. Duan, I. W. Tsang, D. Xu, and S. J. Maybank, “Domain transfer SVM for video concept detection,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2019). IEEE,2019.

[13]Xue, Z., Wei, J., & Guo, W. A Real-Time Naive Bayes Classifier Accelerator on FPGA. IEEE Access,Tianjin , China 2020

[14] Yiqing Guo, Student Member IEEE, Xiuping Jia, Effective Sequential Classifier Training for Multitemporal Remote Sensing Image Classification. IEEE Transactions on Image Processing 2018

[15]Kim, B. S. Park, S.B. Comments on ``An Automated Approach to the Design of Decision Tree Classifiers’’. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8(4), 560–560 .Year 2018

[16]Yilin Bi, Wangdong Jiang, Chuntian Luo,Shengjia Cao,Peng GuoJianjun Zhang "Application of Random Forest Classifier in Loan Default Forecast" Springer 2020

[17]Chen, L., & Tang, H. L. Improved computation of beliefs based on confusion matrix for combining multiple classifiers 2018.

[18]W. Fang, L. Wang and P. Ren, "Tinier-YOLO: A Real-Time Object Detection Method for Constrained Environments," in *IEEE Access*,2019

[19]P. Sun, D. Wang, V. C. Mok and L. Shi, "Comparison of Feature Selection Methods and Machine Learning Classifiers for Radiomics Analysis in Glioma Grading," in *IEEE Access*, vol. 7, pp. 102010-102020, 2019

[20]J. Li, Z. Liu, X. Jia and X. Huang, "Scale adaptive region selection for deblurring," in *The Journal of Engineering*, vol. 2016, no. 9, pp. 318-320, 9 2016

[21]N. Pang, J. Zhang, C. Zhang and X. Qin, "Parallel Hierarchical Subspace Clustering of Categorical Data," in *IEEE Transactions on Computers*, 1 April 2019

[22]X. Zou, Y. Hu, Z. Tian and K. Shen, "Logistic Regression Model Optimization and Case Analysis," 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT), Dalian, China, 2019

[23]S. Zhai, D. Shang, S. Wang and S. Dong, "DF-SSD: An Improved SSD Object Detection Algorithm Based on DenseNet and Feature Fusion," in *IEEE Access*, vol. 8, pp. 24344-24357, 2020

[24]Z. Xiao, P. Xu, X. Wang, L. Chen and F. An, "A Multi-Class Objects Detection Coprocessor With Dual Feature Space and Weighted Softmax," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 9, pp. 1629-1633, Sept. 2020

[25]Q. Mao, H. Sun, Y. Liu and R. Jia, "Mini-YOLOv3: Real-Time Object Detector for Embedded Applications," in *IEEE Access*, vol. 7, pp. 133529-133538, 2019

[26]S. Albahli, N. Nida, A. Irtaza, M. H. Yousaf and M. T. Mahmood, "Melanoma Lesion Detection and Segmentation Using YOLOv4-DarkNet and Active Contour," in *IEEE Access*, vol. 8, pp. 198403-198414, 2020.

[27]K. P. Sinaga and M. Yang, "Unsupervised K-Means Clustering Algorithm," in *IEEE Access*, vol. 8, pp. 80716-80727, 2020

[28]Z. Yu, H. Chen, J. Liu, J. You, H. Leung and G. Han, "Hybrid k -Nearest Neighbor Classifier," in *IEEE Transactions on Cybernetics*, vol. 46, pp. 1263-1275, 2016

[29] Shao, Rui; Lan, Xiangyuan; Yuen, Pong C. *Joint Discriminative Learning of Deep Dynamic Textures for 3D Mask Face Anti-spoofing. IEEE Transactions on Information Forensics and Security, 2018*

[30] Sun, Wenyun; Song, Yu; Chen, Changsheng; Huang, Jiwu; Kot, Alex C. *Face Spoofing Detection Based on Local Ternary Label Supervision in Fully Convolutional Networks. IEEE Transactions on Information Forensics and Security. April* 2020

[31] Yao, Chengtang; Jia, Yunde; Di, Huijun; Wu, Yuwei .*Face Spoofing Detection Using Relativity Representation on Riemannian Manifold. IEEE Transactions on Information Forensics and Security* 2020

[32] N. Alyuz, B. Gokberk and L. Akarun, "3-D Face Recognition Under Occlusion Using Masked Projection," in *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 5, pp. 789-802, May 2013,

[33] S. Chen, W. Liu and G. Zhang, "Efficient Transfer Learning Combined Skip-Connected Structure for Masked Face Poses Classification," in *IEEE Access*, vol. 8, May 2020.

[35] J.Guo, C. Lin, M. Wu, C. Chang and H. Lee, "Complexity Reduced Face Detection Using Probability-Based Face Mask Prefiltering and Pixel-Based Hierarchical-Feature Adaboosting," in *IEEE Signal Processing Letters* Aug. 2018

[36]O. Surinta and T. Khamket, "Recognizing Pornographic Images using Deep Convolutional Neural Networks," *2019 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT-NCON)*, Nan, Thailand, 2019

[37]C. Qin, J. Schlemper, J. Caballero, A. N. Price, J. V. Hajnal and D. Rueckert, "Convolutional Recurrent Neural Networks for Dynamic MR Image Reconstruction," in *IEEE Transactions on Medical Imaging* Jan. 2019

[38] A. Van Opbroek, H. C. Achterberg, M. W. Vernooij and M. De Bruijne, "Transfer Learning for Image Segmentation by Combining Image Weighting and Kernel Learning," in *IEEE Transactions on Medical Imaging*, vol. 38, no. 1, pp. 213-224, Jan. 2019

[39] R. Tennakoon *et al.*, "Classification of Volumetric Images Using Multi-Instance Learning and Extreme Value Theorem," in *IEEE Transactions on Medical Imaging*, vol. 39, no. 4, pp. 854-865, April 2020

[40] C. Yao, Y. Jia, H. Di and Y. Wu, "Face Spoofing Detection Using Relativity Representation on Riemannian Manifold," in *IEEE Transactions on Information Forensics and Security*, 2020

[41] M. J. J. P. van Grinsven, B. van Ginneken, C. B. Hoyng, T. Theelen and C. I. Sánchez, "Fast Convolutional Neural Network Training Using Selective Data Sampling: Application to Hemorrhage Detection in Color Fundus Images," in *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1273-1284, May 2016.

[42] R. Xin, J. Zhang and Y. Shao, "Complex network classification with convolutional neural network," in *Tsinghua Science and Technology*, vol. 25, no. 4, pp. 447-457, Aug. 2020

[43] Y. Ying, J. Su, P. Shan, L. Miao, X. Wang and S. Peng, "Rectified Exponential Units for Convolutional Neural Networks," in *IEEE Access*, vol. 7, pp. 101633-101640, 2019.

[44] F. Guo, R. He and J. Dang, "Implicit Discourse Relation Recognition via a BiLSTM-CNN Architecture With Dynamic Chunk-Based Max Pooling," in *IEEE Access*, vol. 7, pp. 169281-169292, 2019.

[45] J. Dolz, K. Gopinath, J. Yuan, H. Lombaert, C. Desrosiers and I. Ben Ayed, "HyperDense-Net: A Hyper-Densely Connected CNN for Multi-Modal Image Segmentation," in *IEEE Transactions on Medical Imaging*, vol. 38, no. 5, pp. 1116-1126, May 2019.

[46] A. Y. Alanis, "Electricity Prices Forecasting using Artificial Neural Networks," in *IEEE Latin America Transactions*, vol. 16, no. 1, pp. 105-111, Jan. 2018

[47] S. Scardapane, S. Van Vaerenbergh, A. Hussain and A. Uncini, "Complex-Valued Neural Networks With Nonparametric Activation Functions," in *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 2, pp. 140-150, April 2020.

[48] D. -J. Chang, B. -G. Nam and S. -T. Ryu, "Compact Mixed-Signal Convolutional Neural Network Using a Single Modular Neuron," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 12, pp. 5189-5199, Dec. 2020

[49] A. A. Arkadan, Y. Chen, S. Subramanian and S. R. H. Hoole, "NDT identification of a crack using ANNs with stochastic gradient descent," in *IEEE Transactions on Magnetics*, vol. 31, no. 3, pp. May 2018.

[50] D. Chang, M. Lin and C. Zhang, "On the Generalization Ability of Online Gradient Descent Algorithm Under the Quadratic Growth Condition," in *IEEE Transactions on Neural Networks and Learning Systems*, Oct. 2018

[51] T. M. Hoang, G. P. Nam, J. Cho and I. -J. Kim, "DEFace: Deep Efficient Face Network for Small Scale Variations," in *IEEE Access.*2020

[52] G. H. Minari *et al*., "Anomalies Identification in Images from Security Video Cameras Using Mask R-CNN," in *IEEE Latin America Transactions*, vol. 18, no. 03, pp. 530-536, March 2020

[53] J. Zhang, F. Han, Y. Chun and W. Chen, "A Novel Detection Framework About Conditions of Wearing Face Mask for Helping Control the Spread of COVID-19," in *IEEE Access,* 2021

[54] R. Ranjan *et al*., "A Fast and Accurate System for Face Detection, Identification, and Verification," in *IEEE Transactions on Biometrics, Behavior, and Identity Science*, April 2019

arush_mtech

ORIGINALITY REPORT

7%	2%	2%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Jaypee University of Information Technology Student Paper	4%
2	www.mdpi.com Internet Source	1%
3	Submitted to Macquarie University Student Paper	<1%
4	Submitted to Higher Education Commission Pakistan Student Paper	<1%
5	"Computer Vision – ECCV 2018", Springer Science and Business Media LLC, 2018 Publication	<1%
6	"Computational Intelligence: Theories, Applications and Future Directions - Volume II", Springer Science and Business Media LLC, 2019 Publication	<1%
7	Submitted to University of East London Student Paper	<1%

9	Submitted to University of Auckland Student Paper	<1 %
10	analyticsindiamag.com Internet Source	<1 %
11	lineonestudios.com Internet Source	<1 %
12	"Hybrid Artificial Intelligent Systems", Springer Science and Business Media LLC, 2018 Publication	<1 %
13	Zhe Liu, Chunyang Chen, Junjie Wang, Yuekai Huang, Jun Hu, Qing Wang. "Owl eyes", Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering, 2020 Publication	<1 %
14	aditi-mittal.medium.com Internet Source	<1 %
15	"Advances in Neural Networks - ISNN 2017", Springer Science and Business Media LLC, 2017 Publication	<1 %
16	"Neural Information Processing", Springer Science and Business Media LLC, 2017 Publication	<1 %

17	www.alazhar.edu.ps Internet Source	<1 %
18	Atefeh Abdolmanafi, Luc Duong, Nagib Dahdah, Farida Cheriet. "Intra-Slice Motion Correction of Intravascular OCT Images Using Deep Features", IEEE Journal of Biomedical and Health Informatics, 2019 Publication	<1 %

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ……8/07/2021…………………….

Type of Document (Tick): PhD Thesis M.Tech Dissertation/ Report B.Tech Project Report

	M.TECH		**Paper**

Name: ________ARUSH KAUSHAL___________________ __**Department:** ___CSE______________

Enrolment No ___192203______ **Contact No.** _________**9418981032**_____________________**E-mail.**

__arushkaushal@gmail.com_____________________________________ **Name of the Supervisor:**

____________DR. VIVEK SEHGAL____ **Title**

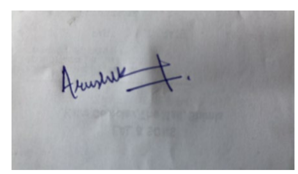
of the Thesis/Dissertation/Project Report/Paper (In Capital letters): ____PARALLEL STUDY OF CNN TRANSFER LEARNING MODEL FOR OBJECT DETECTION____________________

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages = 78
- Total No. of Preliminary pages = 3
- Total No. of pages accommodate bibliography/references = 4



(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ………7………..(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

Vivek Sehgal

(Signature of Guide/Supervisor) Signature of HOD

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	· All Preliminary Pages · Bibliography/Images/Quotes · 14 Words String		Word Counts	
			Character Counts	
Report Generated on		**Submission ID**	Total Pages Scanned	
			File Size	

Checked by
Name & Signature Librarian

………………………………………………………………………………………………………

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com