Educate Elevate Enlighten

JSS Mahavidyapeetha

JSS Science And Technology University

(Established Under JSS Science and Technology University Act No. 43 of 2013)

(Formerly Known as SJCE)

# Department of Computer Science and Engineering (2021-22)

## Project Report

| | |
|---|---|
| **Batch No:** 40 | **Guide Name: Prof. SHWETHASHREE G C** |
| **Project Title: OBJECT DETECTION AND RECOGNITION IN VIDEOS USING DEEP LEARNING** | |

| USN | Student Name |
|---|---|
| 01JST18CS175 | V HARSHITH |
| 01JST18CS159 | VACHAN S PATIL |
| 01JST18CS169 | VIKASH KUMAR |
| 01JST18CS045 | K S MOHITH |

Signature of Guide

(Name of Guide)

Signature of HOD

(Dr. S Srinath)

Educate Elevate Enlighten

JSS Mahavidyapeetha

JSS Science And Technology University

(Established Under JSS Science and Technology University Act No. 43 of 2013)

(Formerly Known as SJCE)

JSS SCIENCE AND TECHNOLOGY UNIVERSITY MYSURU

# Department of Computer Science and Engineering (2021-22)

## Report Evaluation Certificate

| | |
|---|---|
| **Batch No: 40** | **Guide Name: Prof. SHWETHASHREE G C** |
| **Project Title: OBJECT DETECTION AND RECOGNITION IN VIDEOS USING DEEP LEARNING** | |

| USN | Student Name |
|---|---|
| 01JST18CS175 | V HARSHITH |
| 01JST18CS159 | VACHAN PATIL |
| 01JST18CS169 | VIKASH KUMAR |
| 01JST18CS045 | K S MOHITH |

| | |
|---|---|
| Can the project be considered as final year project (yes/no) | |
| Is the problem statement and objectives of the project clear? (yes/no) | |
| Can the project be completed in time? (yes/no) | |
| Does this project have the potential to be converted into a product? (yes/no) | |
| Does this project have social relevance? (yes/no) | |

**Note: if 'No' please write your Remarks/ Suggestions on the backside of this page**

**Evaluation Committee**

**Name** **Signature**

# ABSTRACT

Object detection and recognition are one of the most basic and central tasks in computer vision. Its task is to find all the interesting objects in the image, and determine the category and location of the objects. Object detection and recognition are widely used and have strong practical value and research prospects. In recent years, with the development of convolutional neural networks, significant breakthroughs have been made in object detection. Still Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper when compared to previously used ones.

# ACKNOWLEDGEMENT

It gives us immense pleasure to write an acknowledgement to this project, a contribution of all the people who helped to realize it. We extend our deep regards to Dr. S.B. Kivade, Honorable Principal of JSS Science and Technology University, for providing an excellent environment for our education and his encouragement throughout our stay in college. We would like to convey our heartfelt thanks to our HOD, Dr. S Srinath, for giving us the opportunity to embark upon this topic. We would like to thank our project guide, Prof. Shwethashree G C for their invaluable guidance and enthusiastic assistance and for providing us support and constructive suggestions for the betterment of the project, without which this project would not have been possible. We appreciate the timely help and kind cooperation of our lecturers, other staff members of the department and our seniors, with whom we have come up all the way during our project work without whose support this project would not have been successful. Finally, we would like to thank our friends for providing numerous insightful suggestions. We also convey our sincere thanks to all those who have contributed to this learning opportunity at every step of this project.

# TABLE OF CONTENTS:

# 1. INTRODUCTION

## 1.1 PROBLEM STATEMENT

With the rapid development of modern technology and the popularization of social media and self-media, a large amount of visual information has followed. The images and videos have become an important carriers of a vast network of information. They bring us a convenient way of recording and sharing information about different objects, but it's difficult for us to classify and recognize the gathered information directly. Hence, to end this complication, the use of the computer to classify and recognize data or objects in these images and videos has gained significant prominence.

## 1.2 AIM

To build a web application for implementing a machine learning model that detects and recognizes different objects in images and videos using deep learning.

## 1.3 HYPOTHESIS

The Web application should allow users to pass images and videos of different objects to the backend. It will then be run through the pre-trained model and generate an image or video having objects classified and highlighted in it.

## 1.4 OBJECTIVES

1. To build an effective neural network for detecting and classifying different objects in the image or videos.
2. To be able to upload images or videos to the web application.
3. To be able to identify different characteristics of the uploaded files.
4. To classify and label the objects in the uploaded image or video.
5. To output the image or video having the identified objects highlighted in them.

## 1.5 APPLICATIONS

Considering the huge dataset the machine learning model is trained on and the efficiency of the model it can be utilized for various applications such as:

- License number plate analysis
- People detection in surveillance cameras
- Medical feature detection in Healthcare
- Object detection in Retail

## 1.6 EXISTING SOLUTION METHODS

A wide range of computer vision applications has become available for object detection and tracking. As a result, numerous real-world applications, such as healthcare monitoring, autonomous driving, video surveillance, anomaly detection, or robot vision, are based on deep learning object detection.

Imaging technology has greatly progressed in recent years. Cameras are smaller, cheaper, and of higher quality than ever before. Meanwhile, computing power has dramatically increased and has become much more efficient. In past years, computing platforms moved toward parallelization through multi-core processing, graphical processing units (GPU), and AI accelerators such as tensor processing units (TPU)

Such hardware allows to perform computer vision for object detection and tracking in near real-time implementations. Hence, rapid development in deep convolutional neural networks (CNN) and GPU's enhanced computing power are the main drivers behind the great advancement of computer vision-based object detection.

Object detection can be performed using either traditional (1) image processing techniques or modern (2) deep learning networks.

- Image processing techniques generally don't require historical data for training and are unsupervised.

    - Pros: Hence, those tasks do not require annotated images, where humans labeled data manually (for supervised training).

    - Cons: These techniques are restricted to multiple factors, such as complex scenarios (without unicolor background), occlusion (partially hidden objects), illumination and shadows, and clutter effect.

- Deep Learning methods generally depend on supervised training. The performance is limited by the computation power of GPUs which is rapidly increasing year by year.

    - Pros: Deep learning object detection is significantly more robust to occlusion, complex scenes, and challenging illumination.

    - Cons: A huge amount of training data is required; the process of image annotation is labor-intensive and expensive. For example, labeling 500'000 images to train a custom DL object detection algorithm is considered a small dataset. However, many benchmark datasets (MS COCO, Caltech, KITTI, PASCAL VOC, V5) provide the availability of labeled data.

The field of object detection is not as new as it may seem. Object detection has evolved over the past 20 years. The progress of object detection is usually separated into two separate historical periods (before and after the introduction of Deep Learning):

Before 2014 – Traditional Object Detection period

- Viola-Jones Detector (2001)
- HOG Detector (2006)
- DPM (2008)

After 2014 – Deep Learning Detection period

- RCNN and SPPNet (2014)
- Fast RCNN and Faster RCNN (2015)
- Mask R-CNN (2017)
- G-RCNN (2021)
- YOLO (2016)
- SSD (2016)
- RetinaNet (2017)
- YOLOv3 (2018)

## 1.7 EXISTING FEATURES

The existing features of the current solution which are generally Traditional Object Detection models or Deep Learning Detection models are:

1. Allows users to detect and classify images or video frames.
2. Helps in the identification of different features in an image or video.
3. Builds a network or Machine Learning model to help in the reduction of the training period and detection of multiple images.

## 1.8 LIMITATIONS, PROPOSED SOLUTION METHODS, SCOPE, AND NEW FEATURES

### 1.8.1 LIMITATIONS

- As the dataset consists of images and videos, the size of the dataset is relatively large. Thus training such data requires a high-performance GPU and processor.
- Storage issues due to large datasets consisting of images and videos.
- Improper internet connection.

### 1.8.2 PROPOSED SOLUTION METHODS

Object detection refers to the capability of computer and software systems to locate objects in an image or video and identify each object. Object detection has been widely used for face detection, vehicle detection, pedestrian counting, web images, security systems, and driverless cars. There are many ways object detection can be used as well in many fields of practice.

We plan to use high-resolution images of the COCO dataset to train the neural network to classify different objects. A convolution neural network algorithm is used to improve the ability to classify and recognize two-dimensional images and videos, speed up the convergence of the algorithm, reduce the number of iterations and shorten the training period, and achieve good classification results. The algorithm uses different stages to improve the performance of the network, reducing loss function in the network and improving the accuracy of the feature extraction of the network as mentioned below.

First, a recurrent neural network is introduced into the convolutional neural network, and the deep features of the image are learned in parallel using the convolutional neural network and the recurrent neural network. Recurrent Neural Networks(RNN) are the type of Neural Network where the output from the previous step is fed as input to the current step.

Second, according to the idea of ResNet's skip convolution layer, a new residual module ResNet is constructed. A residual neural network (ResNet) is an artificial neural network (ANN) of a kind that builds on constructs known from pyramidal cells in the cerebral cortex. Residual neural networks do this by utilizing skip connections or shortcuts to jump over some layers.

The COCO dataset is passed through all different stages of the algorithm and is verified by the same dataset. The parameter of the created machine learning model, which comprises the algorithm and neural network, can be changed based on the accuracy rate of the object recognition.

A web application is created to provide an abstraction of the model and algorithm. An unidentified object in an image or video is uploaded into the application. The image or video is passed through the machine learning model for the identification of different objects present in it. The model processes and highlights different objects present in it and labels them, based on the trained neural network and machine learning model. The image or video having classified objects in it can be downloaded from the application.

### 1.8.3 SCOPE

The project is mainly developed for the classification and recognition of objects in visual data (Image or Video). Users can access the classification model created in the backend by using a web application. The web application is developed for providing an easier interface for the developed machine learning model. The machine learning model created is a CNN (Convolutional Neural Network).

### 1.8.4 FEATURES

This software is characterized by the following functionalities:

- Allows users to upload images or videos to the website which have to be classified.
- The Neural network created should be able to identify different characteristics of the uploaded files.

- The Neural Network built should be able to reduce the training time and improve the accuracy rate of classification using the concepts of deep residual networks.
- After the image or video is uploaded, the file should be classified and labeled with objects in it.

## 1.9 TIME SCHEDULED FOR COMPLETION OF PROJECT

In accordance with the college's given schedule and the evaluation phases, the scheduled time required for the project completion is 2-3 months.

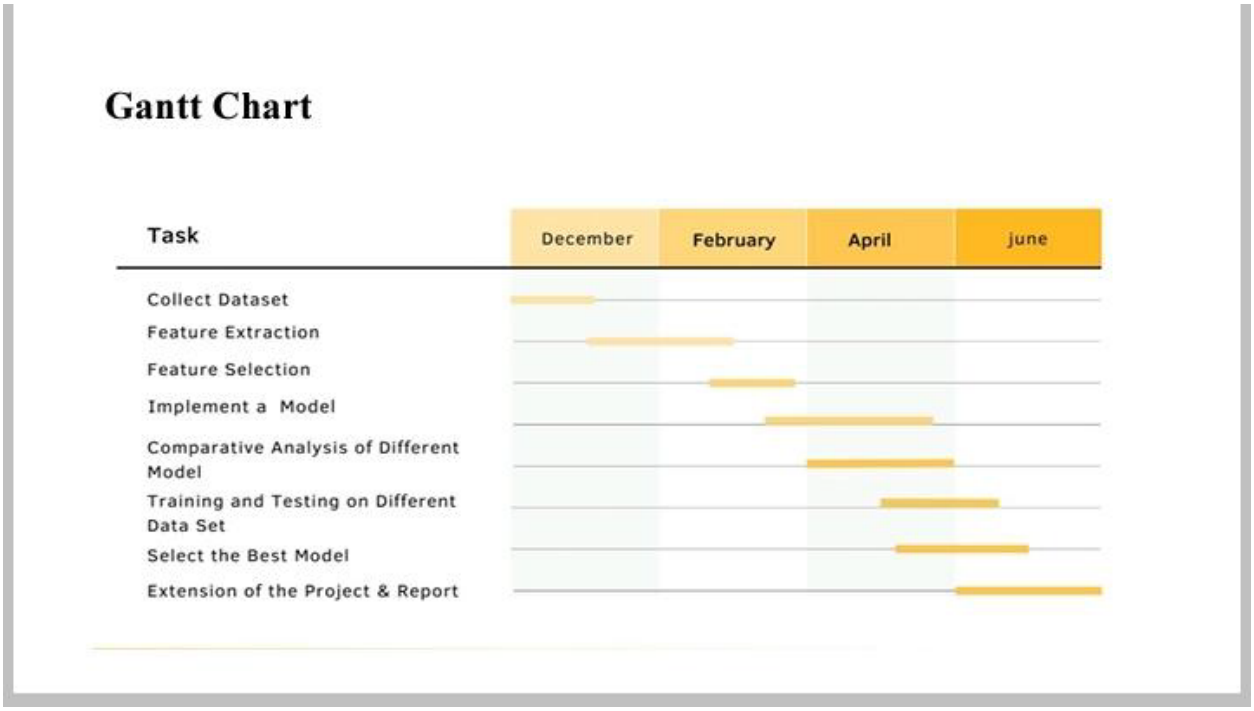



Fig 1.1: Gantt Chart

# 2. LITERATURE SURVEY

## PAPER 2.1: Image Recognition Technology Based on Machine Learning

### By Lijuan Liu, Yanping Wang, and Wanle Chi

### 2.1.1 AIM

Due to the complexity of video images and the distribution of objects in different application backgrounds, classification accuracy becomes important and difficult. Therefore, how to improve the classification method to improve the classification accuracy and classification effect of the image of the ground object is a very meaningful and difficult research topic. With the development of machine learning and the introduction and improvement of various machine learning algorithms, machine learning is of great significance to various application fields in human life. This paper identifies the images based on the machine learning method and classifies the sample using a Neural Network trained by a genetic algorithm.

### 2.1.2 THEME

As an important method in the field of artificial intelligence, machine learning has been widely used in image detection for the classification of various images. Because of its intelligence, good generalization, and high recognition efficiency, it has gradually become the mainstream of image recognition research. In order to complete this experiment, a large amount of target data was collected, but in the field of target recognition, it is very difficult to obtain large-scale effective data. This is also the primary problem that hinders the application of deep learning in the field of image recognition. To this end, it is necessary to find a more effective way to carry out manual data expansion based on the original database, so that deep learning can be effectively applied.

### 2.1.3 APPLICATIONS

As mentioned in the paper this theme has various applications like

1. License number plate analysis.
2. People detection in surveillance cameras etc.

### 2.1.4 DETAILED DISCUSSION OF PAPER

The general pattern recognition system includes three important parts: image preprocessing, feature extraction, and classifier. In traditional image recognition algorithms, they are separated from each other. In the framework of the convolutional neural network, convolution is used to extract features directly, and then the classification results are fed back to the classifier, and the model is jointly optimized by batch gradient descent. The process of computer preprocessing is mainly to separate the image area and background area in the image to

be recognized, refine the image, enhance the image binarization, and improve the speed and efficiency of computer intelligence image recognition post-processing. To restore the authenticity of the image and reduce the false features of the image as much as possible, the unique features of the image can be expressed in numerical form. With the development and progress of technology, the digital image is gradually used in the field of image recognition. The advantages of digital processing technology provide the basis for the further development of image recognition.

Machine learning is a general term for a class of algorithms that attempt to mine the implicit rules from a large amount of historical data and use them for prediction or classification. More specifically, machine learning is looking for a function, and input is sample data. The output is the desired result, but this function is too complicated to be formally expressed. It is important to note that the goal of machine learning is to make the learned functions work well for "new samples," not just for training samples.

The principle of artificial intelligence image recognition technology is the same as that of computer processing data; therefore, simple image data information extraction can be performed by the computer, but when the amount of information is large, the recognition rate of image recognition technology will decrease, and relevant personnel is analyzing The principle of image recognition technology should look for more optimized methods for innovation, to improve the quality and efficiency of image processing. The image recognition technology in artificial intelligence has the advantages of convenience and intelligence. The advantages of the technology directly determine the application quality and effect of image recognition technology in the development of science and technology. First of all, from the perspective of intelligence, the most obvious advantage of artificial intelligence image recognition technology is intelligence. Compared with traditional image processing technology, it shows a clear difference. This function can realize intelligent selection and recognition when processing pictures, such as the face unlocking function in the mobile phone, which is very similar to the intelligent recognition function in image processing, that is, the face unlocking can be permanently used as long as the face unlocking is completed.

## 2.1.5: DESIGN AND IMPLEMENTATION

## 2.1.5.1 TOOLS AND TECHNOLOGY USED

1. OpenCV
2. Python
3. Numpy

## 2.1.5.2 DESIGN

First, information data is obtained. Information collection is a prerequisite for image recognition. It mainly converts various special signals into electrical signals through sensors and

then obtains the required information and data from them. However, the information acquired in image recognition technology belongs to the special data of images. The data must be able to distinguish the gaps between the graphics. Second, information data is preprocessed. The third, is feature extraction and selection. This is the key content of image recognition technology, especially in the recognition mode, the actual operation requirements are higher, which also directly determines whether the image can be successfully recognized and whether the extracted features can be stored. Fourth, classifier design and classification decisions. This is the last step of image recognition. This part mainly formulates the recognition rules according to the operation procedure and recognizes the image according to the standard instead of the chaotic recognition. The purpose is to improve the recognition degree of image processing, thereby improving the efficiency of image evaluation.

### 2.1.5.3 IMPLEMENTATIONS

The labeled subset we collected consists of ten classes of objects with 6000 images in each class. The classes are airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. Since each image in the dataset already comes with a noisy label (the search term used to find the image), all we needed the labelers to do was alter the mislabeled images. To that end, each labeler was given a class and asked to examine all the images which were found with that class as the search term. Since the dataset contains only about 3000 images per search term, the labelers were also asked to examine all the images which were found with a search term that is a hyponym (as defined by WordNet) of the main search term. As an example, some of the hyponyms of a ship are cargo ship, ocean liner, and frigate. The labelers were instructed to reject images that did not belong to their assigned class.

## 2.1.6: ANALYSIS OF THE RESULTS

We compared several methods of classifying the images in the CIFAR-10 dataset. Each of these methods used multinomial logistic regression at its output layer, so we distinguish the methods mainly by the input they took:

1. The raw pixels (unwhitened).

2. The raw pixels (whitened).

3. The features learned by an RBM are trained on the raw pixels.

4. The features learned by an RBM trained on the features learned by the RBM of #3

To complete this experiment, a large amount of target data was collected, but in the field of target recognition, it is very difficult to obtain large-scale effective data. This is also the primary problem that hinders the application of deep learning in the field of image recognition. To this end, it is necessary to find a more effective way to carry out manual data expansion based

on the original database, so that deep learning can be effectively applied. Data in life is ubiquitous, but tagged data is not common. Similarly, it is easier to collect data in the field of image recognition, but manually collecting the collected data is a time-consuming and labor-intensive task. To this end, unsupervised learning algorithms are also the focus of research in deep learning, such as generating confrontational network models. In the correction process of the license plate, this paper mainly focuses on the linear information provided by the framed license plate. If the license plate location module provides a license plate without a frame, then a targeted algorithm should be developed. At the same time, given the control of the generalization accuracy of the classifier in the license plate character recognition, this paper combines the genetic algorithm with the optimal solution search tool which is better than the exhaustive method to solve the global space of the weight of the neural network. After experimental verification, the three solutions with the highest fitness are obtained from the genetic algorithm. The generalization effect after decoding to the neural network is relatively good.

## 2.1.7: CONCLUSION AND FUTURE WORK

Over the years, fields that have used analog imaging for decades have shifted to digital systems, due to their flexibility and affordability. Some of the major fields that rely on such techniques are medicine, photography, feature extraction, remote sensing, computer vision, security monitoring, face detection, and optical character recognition.

- Computer Vision - Computer vision is used in artificial systems to acquire information from images or video signals to then decide the outcome or next action to be taken. Industrial robots and autonomous vehicles depend on such systems to navigate and get their tasks done.
- Face Detection - This method helps in the analysis and matching of integral facial features to help with the detection and identification of faces. Face detection is a type of object class detection and is used extensively in security and surveillance work.
- Video Processing - Much like signal processing, video processing is an important part of digital systems. It is used in television sets, DVDs, and video players to run and display visual data.
- Remote Sensing - Remote sensing uses real-time wireless sensors to gather information about an object at a distance. This technique is used extensively by aircraft, satellites, and ships via ultrasound, Magnetic, or even X-radiation methods.
- Biomedical Analysis - Image processing has found multiple uses in the field of medicine with it being a major source of image diagnosis. Also, it helps in the improvement of techniques such as Computed Tomography and Magnetic Resonance Imaging, helping doctors get better diagnostics, and hence, detect diseases faster.

## 2.1.8: REFERENCES


1. Lijuan Liu, Yanping Wang, and Wanle Chi, College of Computer Science, Neijiang Normal University, Neijiang 641112, Sichuan, INFORMATION CENTER on image recognition based on machine learning.
2. Learning Multiple Layers of Features from Tiny Images Alex Krizhevsky April 8, 2009,
3. Shi X, Chen Z, Wang H, et al. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting", 2015:961-997.
4. Xiao H, Rasul K, Vollgraf R. Fashion-MNIST: "a Novel Image Dataset for Benchmarking Machine Learning Algorithms", 2017:1691-1737.

## PAPER 2.2: Deep Residual Learning for Image Recognition

### By Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun

## ABSTRACT

Deeper neural networks are more difficult to train. This paper presents a residual learning framework to ease the training of networks that are substantially deeper than those used previously. Also, this paper helps explicitly reformulate the layers as learning residual functions concerning the layer inputs, instead of learning unreferenced functions. We are provided with comprehensive empirical evidence that shows that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth.

## 2.2.1 AIM

The main objective of this paper is to the significance of the depth of a neural network. Deep convolutional neural networks have led to a series of breakthroughs in image classification. Deep networks naturally integrate low/mid/high-level features and classifiers in an end-to-end multilayer fashion and the “levels” of features can be enriched by the number of stacked layers (depth). Recent evidence reveals that network depth is of crucial importance, and the leading results on the challenging ImageNet dataset all exploit “very deep” models, with a depth of sixteen to thirty. Many other nontrivial visual recognition tasks have also greatly benefited from very deep models.

## 2.2.2 THEME

The paper’s main theme: is learning better networks as easy as stacking more layers? An obstacle to answering this question was the notorious problem of vanishing/exploding gradients which hamper convergence from the beginning. This problem, however, has been largely addressed by normalized initialization and intermediate normalization layers, which enable networks with tens of layers to start converging for stochastic gradient descent (SGD) with backpropagation. When deeper networks can start converging, a degradation problem has been exposed: with the network depth increasing, accuracy gets saturated (which might be unsurprising) and then degrades rapidly. Unexpectedly, such degradation is not caused by overfitting, and adding more layers to a suitably deep model leads to higher training error, as reported in and thoroughly verified by experiments in paper.

## 2.2.3 APPLICATIONS

ResNet-50 is a convolutional neural network that is 50 layers deep. You can load a pre-trained version of the network trained on more than a million images from the ImageNet database. The pre-trained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature

representations for a wide range of images. The network has an image input size of 224-by-224. This paper gives us various implementations of an effective neural network with low training error. This can also be implemented in various other networks to increase accuracy and training speed.

## 2.2.4: DETAILED DISCUSSION OF PAPER

Deep convolutional neural networks have led to a series of breakthroughs for image classification. Deep networks naturally integrate low/mid/high-level features and classifiers in an end-to-end multilayer fashion, and the "levels" of features can be enriched by the number of stacked layers (depth). Recent evidence reveals that network depth is of crucial importance, and the leading results on the challenging ImageNet dataset all exploit "very deep" models, with a depth of sixteen to thirty. Many other nontrivial visual recognition tasks have also greatly benefited from very deep models.

Driven by the significance of depth, a question arises: Is learning better networks as easy as stacking more layers? An obstacle to answering this question was the notorious problem of vanishing/exploding gradients, which hamper convergence from the beginning. This problem, however, has been largely addressed by normalized initialization and intermediate normalization layers, which enable networks with tens of layers to start converging for stochastic gradient descent (SGD) with backpropagation.

When deeper networks can start converging, a degradation problem has been exposed: with the network depth increasing, accuracy gets saturated (which might be unsurprising) and then degrades rapidly. Unexpectedly, such degradation is not caused by overfitting, and adding more layers to a suitably deep model leads to higher training error, as reported in and thoroughly verified by our experiments. Fig. shows a typical example.

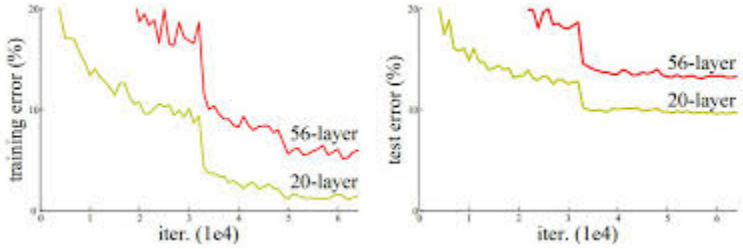



Fig 2.1:- Training and Testing Error

The degradation (of training accuracy) indicates that not all systems are similarly easy to optimize. Let us consider a shallower architecture and its deeper counterpart that adds more layers to it. There exists a solution by construction to the deeper model: the added layers are identity mapping, and the other layers are copied from the learned shallower model. The

existence of this constructed solution indicates that a deeper model should produce no higher training error than its shallower counterpart. But experiments show that our current solvers on hand are unable to find solutions that are comparably good or better than the constructed solution (or unable to do so in infeasible time).

In this paper, we address the degradation problem by introducing a deep residual learning framework. Instead of hoping every few stacked layers directly fit a desired underlying mapping, we explicitly let these layers fit a residual mapping. Formally, denoting the desired underlying mapping as H(x), we let the stacked nonlinear layers fit another mapping of F(x)to:= H(x)□x. The original mapping is recast into F(x)+x. We hypothesize that it is easier to optimize the residual mapping than to optimize the original, unreferenced mapping. To the extreme, if an identity mapping were optimal, it would be easier to push the residual to zero than to fit an identity mapping by a stack of nonlinear layers.

We present comprehensive experiments on ImageNet to show the degradation problem and evaluate our method. We show that: 1) Our extremely deep residual nets are easy to optimize, but the counterpart "plain" nets (that simply stack layers) exhibit higher training error when the depth increases; 2) Our deep residual nets can easily enjoy accuracy gains from greatly increased depth, producing results substantially better than previous networks.

Similar phenomena are also shown on the CIFAR-10 set, suggesting that the optimization difficulties and the effects of our method are not just akin to a particular dataset. We present successfully trained models on this dataset with over 100 layers, and explore models with over 1000 layers.

On the ImageNet classification dataset, we obtain excellent results from extremely deep residual nets. Our 152- layer residual net is the deepest network ever presented on ImageNet, while still having lower complexity than VGG nets. Our ensemble has 3.57% top-5 error on the ImageNet test set and won 1st place in the ILSVRC 2015 classification competition. The extremely deep representations also have excellent generalization performance on other recognition tasks and lead us to further win 1st place on ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation in ILSVRC & COCO 2015 competitions. This strong evidence shows that the residual learning principle is generic, and we expect that it is applicable in other vision and non-vision problems.

## 2.2.5: DESIGN AND IMPLEMENTATION

## 2.2.5.1 TOOLS AND TECHNOLOGY USED

The model proposed in the paper was implemented using

1. Python 3.7
2. TensorFlow 1.13

3. OpenCV
4. Keras
5. ImageAI modules for python
6. Google's open-source ML model training and deployment platform, Collaboratory
7. 4GB of RAM running Windows 10

## 2.2.5.2 DESIGN

We have tested various plain/residual nets, and have observed consistent phenomena. To provide instances for discussion, we describe two models for ImageNet as follows.

**Plain Network**. Our plan baselines are mainly inspired by the philosophy of VGG nets. The convolutional layers mostly have 33 filters and follow two simple design rules: (i) for the same output feature map size, the layers have the same number of filters; and (ii) if the feature map size is halved, the number of filters is doubled to preserve the time complexity per layer. We perform downsampling directly by convolutional layers that have a stride of 2. The network ends with a global average pooling layer and a 1000-way fully-connected layer with softmax. It is worth noticing that our model has fewer filters and lower complexity than VGG nets. Our 34-layer baseline has 3.6 billion FLOPs (multiply-adds), which is only 18% of VGG-19 (19.6 billion FLOPs).

Residual Network. Based on the above plain network, we insert shortcut connections that turn the network into its counterpart residual version. The identity shortcuts can be directly used when the input and output are of the same dimensions. When the dimensions increase, we consider two options: (A) The shortcut still performs identity mapping, with extra zero entries padded for increasing dimensions. This option introduces no extra parameter; (B) The projection shortcut in Eqn. (2) is used to match dimensions (done by 11 convolutions). For both options, when the shortcuts go across feature maps of two sizes, they have performed with a stride of 2.

## 2.2.5.3 IMPLEMENTATIONS

Our implementation for ImageNet follows the practice in The image is resized with its shorter side randomly sampled in for scale augmentation. A 224x224 crop is randomly sampled from an image or its horizontal flip, with the per-pixel mean subtracted. The standard color augmentation is used. We adopt batch normalization (BN) right after each convolution and before activation, following. We initialize the weights as in and train all plain/residual nets from scratch. We use SGD with a mini-batch size of 256. The learning rate starts from 0.1 and is divided by 10 when the error plateaus and the models are trained for up to 60104 iterations. We use a weight decay of 0.0001 and a momentum of 0.9.

In testing, for comparison studies, we adopt the standard 10-crop testing. For best results, we adopt the fully convolutional form as in, and average the scores at multiple scales (images are resized such that the shorter side is in f224; 256; 384; 480; 640g).

## 2.2.6: ANALYSIS OF THE RESULTS

We have three major observations from Table. First, the situation is reversed with residual learning – the 34-layer ResNet is better than the 18-layer ResNet (by 2.8%). More importantly, the 34-layer ResNet exhibits considerably lower training error and is generalizable to the validation data. This indicates that the degradation problem is well addressed in this setting and we manage to obtain accuracy gains from increased depth.

| | plain | ResNet |
|---|---|---|
| 18 layers | 27.94 | 27.88 |
| 34 layers | 28.54 | **25.03** |

Table 2.1:- Comparison between Plain Network and Traditional Network

Second, compared to its plain counterpart, the 34-layer ResNet reduces the top-1 error by 3.5%, resulting from the successfully reduced training error. This comparison verifies the effectiveness of residual learning on extremely deep systems

| method | top-1 err. | top-5 err. |
|---|---|---|
| VGG [41] (ILSVRC'14) | - | 8.43† |
| GoogLeNet [44] (ILSVRC'14) | - | 7.89 |
| VGG [41] (v5) | 24.4 | 7.1 |
| PReLU-net [13] | 21.59 | 5.71 |
| BN-inception [16] | 21.99 | 5.81 |
| ResNet-34 B | 21.84 | 5.71 |
| ResNet-34 C | 21.53 | 5.60 |
| ResNet-50 | 20.74 | 5.25 |
| ResNet-101 | 19.87 | 4.60 |
| ResNet-152 | **19.38** | **4.49** |

Table 2.2:- Comparison of Top Errors between Different Networks

Last, we also note that the 18-layer plain/residual nets are comparably accurate, but the 18-layer ResNet converges faster (Fig. right vs. left). When the net is "not overly deep" (18 layers here), the current SGD solver is still able to find good solutions to the plain net. In this case, the ResNet eases the optimization by providing faster convergence at the early stage.

## 2.2.7: CONCLUSION AND FUTURE WORK

To summarize, we can say that the skip connection introduced in ResNet architecture has helped a lot to increase the performance of the neural network with a large number of layers. ResNets are basically like different networks with slight modifications. The architecture involves

the same functional steps as in CNN or others but an additional step is introduced to tackle certain issues like vanishing gradient problems etc.

The report also recommends that future work can work on the multi-label classification for structural images. The classification is divided into three different tasks. However, one structural image always has multiple attributes. For instance, one pixel-level image can be classified as non-collapse and minor damage. Therefore, if future research can build one single multi-label classification model, the model can be more practical for post-earthquake reconnaissance.

## 2.2.8: REFERENCES


1. S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun. Object detection networks on convolutional feature maps. arXiv:1504.06066, 2017.
2. R. Girshick. Fast R-CNN. In ICCV, 2016
3. J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation.

## PAPER 2.3: New Generation Deep Learning For Video Object Detection

**By Licheng Jiao, Ruohan Zhang, Fang Liu**

## ABSTRACT

Video object detection, a basic task in the computer vision field, is rapidly evolving and widely used. In recent years, deep learning methods have rapidly become widespread in the field of video object detection, achieving excellent results compared with those of traditional methods. However, the presence of duplicate information and abundant spatiotemporal information in video data poses a serious challenge to video object detection. Therefore, in recent years, many scholars have investigated deep learning detection algorithms in the context of video data and have achieved remarkable results.

### 2.3.1 AIM

Due to the complexity of video images and the distribution of objects in different application backgrounds, classification accuracy becomes important and difficult. Therefore, how to improve the classification method to improve the classification accuracy and classification effect of the image of the ground object is very meaningful and difficult research in CNN. With the development of machine learning and the introduction and improvement of various machine learning algorithms, machine learning is of great significance to various application fields in human life. This paper identifies the images based on the machine learning method and classifies the sample using a Neural Network trained by a genetic algorithm.

### 2.3.2 THEME

The main theme of this paper is to show how there is a shift in the usage of various algorithms. In traditional object detection, the histogram of oriented gradients (HOG), the scale-invariant feature transform (SIFT), the frame difference (FD) method, and the optical flow method is used to detect objects in the video. Traditional algorithms cannot meet the requirements for video data analysis with precision. With the development of deep learning, deep convolutional neural networks (CNNs) have been extensively applied to the field of object detection, and considerable progress has been made compared with traditional methods.

### 2.3.3 APPLICATIONS

Video object detection algorithms are required in numerous application scenarios, such as driver-less technology, intelligent video surveillance, and robot navigation. Object detection is breaking into a wide scope of enterprises, with use cases extending from individual security to efficiency in the working environment. Object detection is applied in numerous territories of

image processing, including picture retrieval, security, observation, computerized vehicle systems, and machine investigation. Critical difficulties remain in the field of object detection. The potential outcomes are inestimable in future use cases for object detection.

### 2.3.4 : DETAILED DISCUSSION OF PAPER

In this paper, we have been introduced to the current state-of-the-art methods for video object detection from the perspective of their adopted solution strategies, and we employed representative video object detection methods and their related works to explain and summarize the various approaches. In essence, we described the difficulties and strategies used by video object detection methods to date. The article is divided into four sections based on the solution approach taken by the methods, from the most direct postprocessing method to the introduction of additional models, the feature filtering mechanism, and, finally, some interesting and effective networks. Video object detection is both an important research topic and a complex problem in practical applications. As video object detection is a popular and promising field in the field of machine learning, multiple object detection algorithms have been established, and the demands for video data processing have gradually increased. At present, the computational loads of the existing algorithms and the ability to achieve real-time performance levels must be addressed because they restrict the applications of these algorithms. However, as detection algorithms become increasingly mature, we believe that superior methods will be developed to resolve these problems.

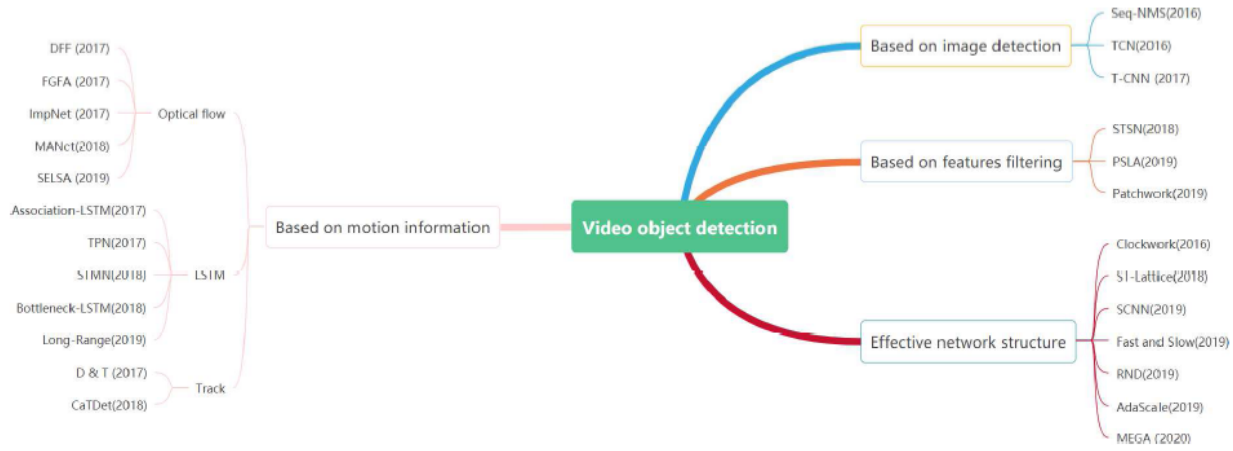



Fig 2.2: Different Video and Image Object Detection Algorithms

**FEATURE FILTERING FOR VIDEO OBJECT DETECTION:**

The feature filtering mechanism is the result of the human brain. After quickly browsing an image, the human brain can focus on certain important areas while suppressing useless information and, thus, can obtain the desired results with relatively little analysis. The feature filtering mechanism of a neural network is the same, that is, a neural network filter features focus

on relatively critical features, and the suppressor discards unnecessary calculations. This method of mimicking the human visual feature filtering mechanism has greatly improved the efficiency and accuracy of visual information processing, such as the attention mechanism. Many tasks in computer vision affect the final model performance due to insufficient semantic information. The widespread application of the feature filtering mechanism further illustrates its excellent performance. However, at present, most feature filtering models focus on only one image. The feature filtering model can extract significant features for the key parts of images, which is very beneficial for video object detection. The role of the feature filtering mechanism in video object detection is quite clear. The feature filtering mechanism can filter the features of a video frame, select relatively representative features, propagate key features for detection and delineate the key areas that deserve feature filtering in subsequent frames. For the spatiotemporal feature filtering mechanism, it is possible to propagate important spatiotemporal information and ignore redundant information. The feature filtering mechanism is very useful for video detection tasks.

**EFFICIENT STRUCTURE FOR NEURAL NETWORK:**

In addition to the strategies mentioned above, some non-mainstream methods are explorations on video object detection tasks. These methods analyze video data from different aspects and try to use more innovative and effective methods to solve video object detection tasks, such as the statistical convolutional method, clockwork convnets, and relation distillation networks (RDNs). Although these methods are diverse, they have common features: they reduce redundant calculations and implement the communication of features between video frames to enhance weak features. These methods comprise highly efficient architectures in the video domain.

CNN is a novel algorithm that represents the video data distribution through independent component analysis (ICA), thereby capturing its temporal and contextual correlations. The method considers a set of video data as a large input, and multiple frames are input at a time to calculate the video data distribution. The “sum” and “max” operations of the neural network are redefined, but all necessary CNN operations (such as convolution, rectified linear unit (ReLU), and batch normalization) are retained. These operations are calculated without using deterministic numbers; instead, parameterized statistical distributions are employed. Therefore, SCNN can be perfectly integrated with a neural network. This strategy allows SCNN to efficiently process multiple frames of related images, accelerating existing CNN models significantly. The work performed with SCNN is truly novel because this network processes the images of multiple frames at the same time, greatly improving the detection speed. Consequently, although its accuracy is not yet satisfactory, this method has promising prospects.

## 2.3.5: DESIGN AND IMPLEMENTATION

## 2.3.5.1 TOOLS AND TECHNOLOGY USED

1. Python
2. Numpy
3. Matplotlib
4. Pandas
5. OpenCV

## 2.3.5.2 DESIGN

**1) Importing necessary modules**

There are many ways of implementing real-time video object detection. This process begins by initially importing all the necessary modules like OpenCV, TensorFlow, etc. The modules are used to generate a machine learning model, feed the camera feed to the model, and identify objects present.

**2) Generating a model**

There are many machine learning models and neural networks that can be used for the implementation of video object detection. In this paper, they have shown many differences of using an efficient neural network for video object detection. One of these methods is using a pre-trained Caffe model called MobileNet SSD. A pre-trained model which was initially created is loaded into the machine and can be used to perform classification though the accuracy depends on the datasets and the model itself.

**3) Sending a camera feed into the model**

The live video footage is fed from the camera into the model. The model then tries to detect multiple objects in the frame based on its confidence value. The camera footage is passed by using an OpenCV module.

**4) Detecting objects from the camera footage**

The camera footage is fed into the model. The model processes all objects and generates a precision point (confidence) which is used to determine which object is which. The model has some threshold value based on which it detects objects. The model was initially trained using certain images of different classes. This model constructed on the training set determines the objects according to the trained classes. The model does have a few limitations of not detecting the objects with lesser precision or giving the wrong detection but the main objective of this model is the speed aspect of the MobileNet SSD.

**5) Displaying the detected objects**

After the detection of the objects, the classified object is differentiated by the colored square and the name of the object on the left corner of the square.

### 2.3.5.3 IMPLEMENTATIONS

In the implementation of this paper, we have considered a pre-trained Caffe model of MobileNet SSD. We pass live footage through our camera into this model which shows different objects in that particular frame. MobileNet uses depth-wise separable convolutions. It significantly reduces the number of parameters when compared to the network with regular convolutions with the same depth in the nets. This results in lightweight deep neural networks.

MobileNet is a class of CNN that was open-sourced by Google, and therefore, this gives us an excellent starting point for training our classifiers that are insanely small and insanely fast.

## 2.3.6: ANALYSIS OF THE RESULTS

The model firstly takes a plain image or the video frame without any detected object. The image is passed through the MobileNet Caffe Model. In the model, various classes of different objects are defined. The model classifies accordingly based on any image.

Fig 2.3: Plain Image

After the image is passed through the model the objects which are present are detected and a colored square is created for each object. This model also gives the precision of the detected image which is used for further analysis or any modification. The color of each square depends on the different images classified.

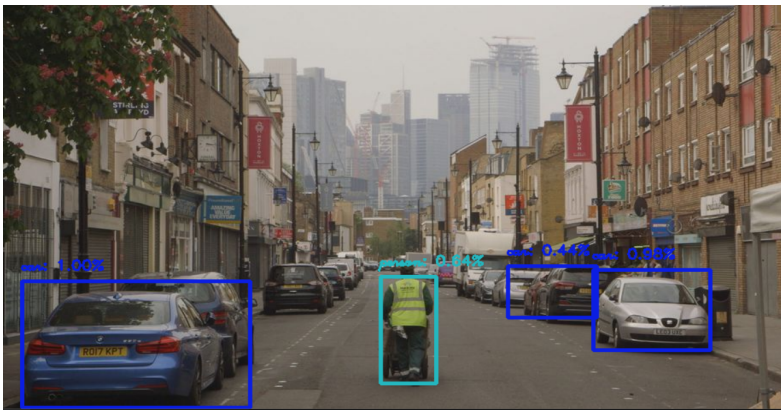


Fig 2.4: Classified Image

## 2.3.7: CONCLUSION AND FUTURE WORK

In this paper, we are introduced to the current state-of-the-art methods for video object detection from the perspective of their adopted solution strategies, and we employed representative video object detection methods and their related works to explain and summarize the various approaches. In essence, we described the difficulties and strategies used by video object detection methods to date.

Video object detection is both an important research topic and a complex problem in practical applications. As video object detection is a popular and promising field in the field of machine learning, multiple object detection algorithms have been established, and the demands for video data processing have gradually increased. At present, the computational loads of the existing algorithms and the ability to achieve real-time performance levels must be addressed because they restrict the applications of these algorithms. However, as detection algorithms become increasingly mature, we believe that superior methods will be developed to resolve these problems.

## 2.3.8: REFERENCES


1. L. Jiao, S. Yang, F. Liu, S. Wang, and Z. Feng, “Seventy years of neural networks: Review and prospect,” Chin. J. Comput., vol. 39, no. 8, pp. 1697–1717, Aug. 2016
2. L. Jiao et al., “A survey of deep learning-based object detection,” IEEEAccess, vol. 7, pp. 128837–128868, 2019
3. R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., June. 2014, pp. 580–587

## PAPER 2.4: You Only Look Once: Unified, Real-Time Object Detection

### By Joseph Redmon, Santosh Divvala, Ross Girshick

## ABSTRACT

YOLO is a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, here we frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance. This makes the unified architecture extremely fast. The base YOLO model processes images in real-time at 45 frames per second. Compared to state-of-the-art detection systems, YOLO makes more localization errors but is less likely to predict false positives in the background. Finally, YOLO learns very general representations of objects. It outperforms other detection methods, including DPM and R-CNN, when generalizing from natural images to other domains like artwork.

## 2.4.1 AIM

The main objective of this paper is to understand and implement You Only Look Once for unified real-time object detection. Deep convolutional neural networks have led to a series of breakthroughs in image classification. Deep networks naturally integrate low/mid/high-level features and classifiers in an end-to-end multilayer fashion and the “levels” of features can be enriched by the number of stacked layers (depth). Recent evidence reveals that network depth is of crucial importance, and the leading results on the challenging ImageNet dataset all exploit “very deep” models, with a depth of sixteen to thirty. Many other nontrivial visual recognition tasks have also greatly benefited from very deep models.

## 2.4.2 THEME

Humans glance at an image and instantly know what objects are in the image, where they are, and how they interact. The human visual system is fast and accurate, allowing us to perform complex tasks like driving with little conscious thought. Fast, accurate algorithms for object detection would allow computers to drive cars without specialized sensors, enable assistive devices to convey real-time scene information to human users, and unlock the potential for general-purpose, responsive robotic systems. Current detection systems repurpose classifiers to perform detection. To detect an object, these systems take a classifier for that object and evaluate it at various locations, and scale it into a test image. Systems like deformable parts models (DPM) use a sliding window approach where the classifier is run at evenly spaced locations over the entire image. In this paper, we reframe object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. Using our

system, you only look once (YOLO) at an image to predict what objects are present and where they are.

## 2.4.3 APPLICATIONS

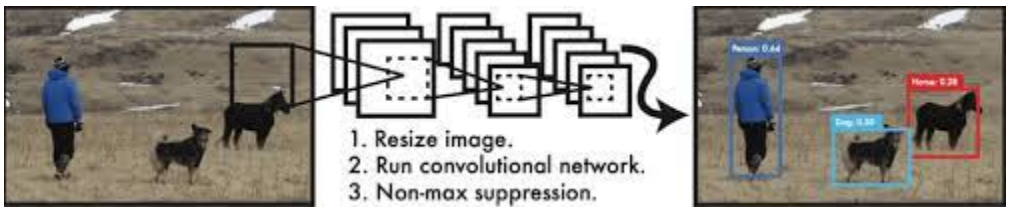
The YOLO algorithm can be used in autonomous cars to detect objects around cars such as vehicles, people, and parking signals. Object detection in autonomous cars is done to avoid collision since no human driver is controlling the car. This algorithm is used to detect various types of animals in forests. This type of detection is used by wildlife rangers and journalists to identify animals in videos (both recorded and real-time) and images.

## 2.4.4: DETAILED DISCUSSION OF PAPER

More recent approaches like R-CNN use region proposal methods to first generate potential bounding boxes in an image and then run a classifier on these proposed boxes. After classification, post-processing is used to refine the bounding boxes, eliminate duplicate detections, and rescore the boxes based on other objects in the scene. Those complex pipelines are slow and hard to optimize because each component must be trained separately.

We reframe object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. Using our system, you only look once (YOLO) at an image to predict what objects are present and where they are.

YOLO is refreshingly simple: as seen in the below figure. A single convolutional network simultaneously predicts multiple bounding boxes and class probabilities for those boxes. YOLO trains on full images and directly optimizes detection performance. This unified model has several benefits over traditional methods of object detection.




First, YOLO is extremely fast. Since we frame detection as a regression problem we don't need complex pipelines. We simply run our neural network on a new image at test time to predict detections. Our base network runs at 45 frames per second with no batch processing on a Titan X GPU and a fast version runs at more than 150 fps. This means we can process streaming video in real-time with less than 25 milliseconds of latency. Furthermore, YOLO achieves more than twice the mean average precision of other real-time systems.

Second, YOLO reasons globally about the image when making predictions. Unlike sliding window and region proposal-based techniques, YOLO sees the entire image during

training and test time so it implicitly encodes contextual information about classes as well as their appearance. Fast R-CNN, a top detection method, mistakes background patches in an image for objects because it can't see the larger context. YOLO makes less than half the number of background errors compared to Fast R-CNN.

Third, YOLO learns generalizable representations of objects. When trained on natural images and tested on the artwork, YOLO outperforms top detection methods like DPM and R-CNN by a wide margin. Since YOLO is highly generalizable it is less likely to break down when applied to new domains or unexpected inputs.

YOLO still lags behind state-of-the-art detection systems' inaccuracy. While it can quickly identify objects in images it struggles to precisely localize some objects, especially small ones. We examine these tradeoffs further in our experiments.

## 2.4.5 : DESIGN AND IMPLEMENTATION

### 2.4.5.1 TOOLS AND TECHNOLOGY USED

The model proposed in the paper was implemented using

1. Python 3.7
2. TensorFlow 2
3. OpenCV
4. Keras
5. PyTorch
6. Jupyter Notebook

### 2.4.5.2 DESIGN

YOLO is refreshingly simple. A single convolutional network simultaneously predicts multiple bounding boxes and class probabilities for those boxes. YOLO trains on full images and directly optimizes detection performance. This unified model has several benefits over traditional methods of object detection. First, YOLO is extremely fast. Since we frame detection as a regression problem we don't need a complex pipeline. We simply run our neural network on a new image at test time to predict detections. Our base network runs at 45 frames per second with no batch processing on a Titan X GPU and a fast version runs at more than 150 fps. This means we can process streaming video in real-time with less than 25 milliseconds of latency. Furthermore, YOLO achieves more than twice the mean average precision of other real-time systems.

Second, YOLO reasons globally about the image when making predictions. Unlike sliding window and region proposal-based techniques, YOLO sees the entire image during training and test time so it implicitly encodes contextual information about classes as well as their appearance. Fast R-CNN, a top detection method, mistakes background patches in an image for objects because it can't see the larger context. YOLO makes less than half the number of background errors compared to Fast R-CNN.

Third, YOLO learns generalizable representations of objects. When trained on natural images and tested on the artwork, YOLO outperforms top detection methods like DPM and R-CNN by a wide margin. Since YOLO is highly generalizable it is less likely to break down when applied to new domains or unexpected inputs. YOLO still lags behind state-of-the-art detection systems' inaccuracy. While it can quickly identify objects in images it struggles to precisely localize some objects, especially small ones.

### 2.4.5.3 IMPLEMENTATIONS

Just like in training, predicting detections for a test image only requires one network evaluation. On PASCAL VOC the network predicts 98 bounding boxes per image and class probabilities for each box. YOLO is extremely fast at test time since it only requires a single network evaluation, unlike classifier-based methods.

The grid design enforces spatial diversity in the bounding box predictions. Often it is clear which grid cell an object falls into and the network only predicts one box for each object. However, some large objects or objects near the border of multiple cells can be well localized by multiple cells. Non-maximal suppression can be used to fix these multiple detections. While not critical to performance as it is for R-CNN or DPM, non-maximal suppression adds 2- 3% in mAP.

## 2.4.6 : ANALYSIS OF THE RESULTS

Many research efforts in object detection focus on making standard detection pipelines fast. However, only Sadeghi et al. produce a detection system that runs in real-time (30 frames per second or better). We compare YOLO to their GPU implementation of DPM which runs either at 30Hz or 100Hz. While the other efforts don't reach the real-time milestone we also compare their relative mAP and speed to examine the accuracy-performance tradeoffs available in object detection systems.

Fast YOLO is the fastest object detection method onCNNSCAL; as far as we know, it is the fastest extant object detector. With 52.7% mAP, it is more than twice as accurate as prior work on real-time detection. YOLO pushes mAP to 63.4% while still maintaining real-time performance.

We also train YOLO using VGG-16. This model is more accurate but also significantly slower than YOLO. It is useful for comparison to other detection systems that rely on VGG-16 but since it is slower than real-time the rest of the paper focuses on our faster models.

| Real-Time Detectors | Train | mAP | FPS |
|---|---|---|---|
| 100Hz DPM [31] | 2007 | 16.0 | 100 |
| 30Hz DPM [31] | 2007 | 26.1 | 30 |
| Fast YOLO | 2007+2012 | 52.7 | **155** |
| YOLO | 2007+2012 | **63.4** | 45 |
| Less Than Real-Time | | | |
| Fastest DPM [38] | 2007 | 30.4 | 15 |
| R-CNN Minus R [20] | 2007 | 53.5 | 6 |
| Fast R-CNN [14] | 2007+2012 | 70.0 | 0.5 |
| Faster R-CNN VGG-16[28] | 2007+2012 | 73.2 | 7 |
| Faster R-CNN ZF [28] | 2007+2012 | 62.1 | 18 |
| YOLO VGG-16 | 2007+2012 | 66.4 | 21 |

Table : Real-Time Systems on PASCAL VOC 2007.

Fast R-CNN speeds up the classification stage of R-CNN but it still relies on selective search which can take around 2 seconds per image to generate bounding box proposals. Thus it has a high mAP but at 0.5 fps it is still far from realtime.

The recent Faster R-CNN replaces selective search with a neural network to propose bounding boxes, similar to Szegedy et al. In our tests, their most accurate model achieves 7 fps while a smaller, less accurate one runs at 18 fps. The VGG-16 version of Faster R-CNN is 10 mAP higher but is also 6 times slower than YOLO. The Zeiler- Fergus Faster R-CNN is only 2.5 times slower than YOLO but is also less accurate.

## 2.4.7 : CONCLUSION AND FUTURE WORK

YOLO is introduced as a unified model for object detection. It is simple to construct and can be trained directly on full images. Unlike classifier-based approaches, YOLO is trained on a loss function that directly corresponds to detection performance and the entire model is trained jointly. Fast YOLO is the fastest general-purpose object detector in the literature and YOLO pushes the state-of-the-art in real-time object detection. YOLO also generalizes well to new domains making it ideal for applications that rely on fast, robust object detection.

Still, YOLO falls short for fast-moving objects and also if there are multiple objects very near or covering each other. It also finds it very difficult to detect small objects or if there are partial images of objects in the photo. There is a great scope of improvement on the YOLO model which can surely be much more enhanced for real-time object detection with extra precision and a faster rate of motion.

## 2.4.8: REFERENCES


1. M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision, 111(1):98–136, Jan. 2015.
2. Z. Shen and X. Xue. Do more dropouts in pool5 feature maps for better object detection. arXiv preprint arXiv:1409.6911, 2014.
3. S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497, 2015
4. Z. Shen and X. Xue. Do more dropouts in pool5 feature maps for better object detection. arXiv preprint arXiv:1409.6911, 2014.

# 3. SYSTEM REQUIREMENTS AND ANALYSIS

## 3.1 EXTERNAL INTERFACE REQUIREMENTS

### 3.1.1 USER INTERFACES:

The software developed is accessible to the end-user as a website, which is responsive allowing the user to run it on all devices, without hampering the user experience, it follows all the website accessibility guidelines and is protected by an SSL certificate. A user will be able to input an image dataset or video frame. The output will object recognition and classification for the given input.

### 3.1.2 HARDWARE INTERFACES:

To access this software the user needs to possess a hardware device with the below-listed configuration:

| | |
|---|---|
| CPU Power | 1.00 GHz and above |
| RAM size(CPU) | 2 GigaBytes and above |
| RAM size(Mobile Phones, Tablets) | 1 GigaByte and above |
| Internet Download Speed | 1.8 Mbps and above |
| Internet Upload Speed | 3.2 Mbps and above |
| Browser version(Firefox) | 95.0 and above |
| Browser version(Google Chrome) | 96.0.4664.110 and above |
| Browser version(Safari) | 15.0 and above |
| Browser version(Microsoft Edge) | 96.0.1054.62 and above |

### 3.1.3 SOFTWARE INTERFACES:

The software is accessible to all users connected to an active internet connection and has the latest version of python and javascript enabled in the browser. Required libraries should be installed on the development environments like Pycharm, and Google Colab Jupyter notebook if used. Opencv is used to process video.

## 3.2 FUNCTIONAL REQUIREMENTS:

In this section, we elaborate on the main features with required models that are essential for data processing and the desired output of these features.

### 3.2.1 OBJECT REPRESENTATION:

How should objects be represented in the model database? What are the important attributes or features of objects that must be captured in these models? For some objects, geometric descriptions may be available and may also be efficient, while for another class one may have to rely on generic or functional features. The representation of an object should capture all relevant information without any redundancies.

### 3.2.2 FEATURE EXTRACTION:

Which features should be detected, and how can they be detected reliably? Most features can be computed in two-dimensional images but they are related to the three-dimensional characteristics of objects. Due to the nature of the image formation process, some features are easy to compute reliably while others are very difficult.

### 3.2.3 FEATURE-MODEL MATCHING:

How can features in images be matched to models in the database? In most object recognition tasks, there are many features and numerous objects. An exhaustive matching approach will solve the recognition problem but may be too slow to be useful. The effectiveness of features and efficiency of a matching technique must be considered in developing a matching approach.

### 3.2.4 HYPOTHESIS FORMING:

How can a set of likely objects based on the feature matching be selected, and how can probabilities be assigned to each possible object? The hypothesis formation step is a heuristic to reduce the size of the search space. This step uses knowledge of the application domain to assign some kind of probability or confidence measure to different objects in the domain.

### 3.2.5 OBJECT VERIFICATION:

How can object models be used to select the most likely object from the set of probable objects in a given image? The presence of each likely object can be verified by using their models. One must examine each plausible hypothesis to verify the presence of the object or ignore it. If the models are geometric, it is easy to precisely verify objects using camera location and other scene parameters. In other cases, it may not be possible to verify the hypothesis.

## 3.3 OTHER-FUNCTIONAL REQUIREMENTS

### 3.3.1 PERFORMANCE REQUIREMENTS

For better accuracy and quality of output, good quality hardware specifications are required. Since our dataset also consists of videos along with images, it needs large storage space and to download it needs High-speed internet connectivity along with a good CPU and GPU will be required for processing image or video data and performing event detection activities. Any computer or mobile device of recent years with a high-speed internet connection, good GPU & CPU is enough to use the product. Comparative analysis is done to constantly improve the performance of the product with the other product in the market.

### 3.3.2 SAFETY REQUIREMENTS

The system is to be safely hosted in a cloud-driven environment with a Secure Sockets Layer(SSL) certificate which is a digital certificate that authenticates a web site's identity and enables an encrypted socket connection between the client and server.

### 3.3.3 SECURITY REQUIREMENTS

The user information is kept private and inaccessible for other user classes to ensure user privacy.

### 3.3.4 SOFTWARE QUALITY ATTRIBUTES

● **Availability**
The software module functions 24x7 and should always be available for the user to avail of services.

● **Correctness**
The software module should be able to deliver the services as intended despite unpredictable interruptions.

● **Usability**
The software must be user-friendly and easy to use without compromising on the requirements.

● **Maintainability**
Administrators must maintain the server efficiently to get an optimal action-responses environment.

### 3.3.5 BUSINESS RULE

The system permits everyone with a mobile phone or a computer to access the website, thus assisting people who are hard of hearing. Thus the revenue generated from our model can further be utilized to enhance the dataset by collecting large amounts of data on which the model is trained, thus making the model more accurate and robust.

# 4. TOOLS AND TECHNOLOGIES

## 4.1: Platforms:

- Google Colab: For collaborating with the team and working to create the customized machine learning model for our use
- Amazon Web Services: For training and testing the machine learning model and also for hosting the website
- Git, Github : it is used for versioning the code and to collaborate with the team mates while developing the code for the website.
- Jupyter Notebook: it is used to work on python code while developing the machine learning code.

## 4.2: Programming Language and Frameworks:

- Python: It is used for creating the machine learning algorithm and also to create scripts to run the model on passed data
- JavaScript, TypeScript: It is used for implementing the website as it will appear to the user in the front end as well as implementing some of the backend capabilities.
- HTML, CSS, Bootstrap: It is used mainly for creating a responsive website where users can see the results.
- Angular: It is used for making the frontend of the website and for integrating it with the backend.
- Node: It is used for creating the backend of the website where a variety of tasks will be performed.

## 4.3: Python modules:

- Keras: It is used for working with the deep learning model
- Tensorflow: Keras works over Tensorflow 2 and it is used for working with neural network
- Numpy: It is used for working with array data.
- OpenCV: It is used for working with images.
- Matplotlib, Plotly, Pillow: It is used for creating charts and graphs.

## 4.4: Operating system:

- Windows 10: The entire application along with the website and the python code machine learning model will be running on a machine with windows 10 as its operating system.

# 5. SYSTEM DESIGN

## 5.1: Architecture and Flow diagrams:

### 5.1.1 Data Flow:

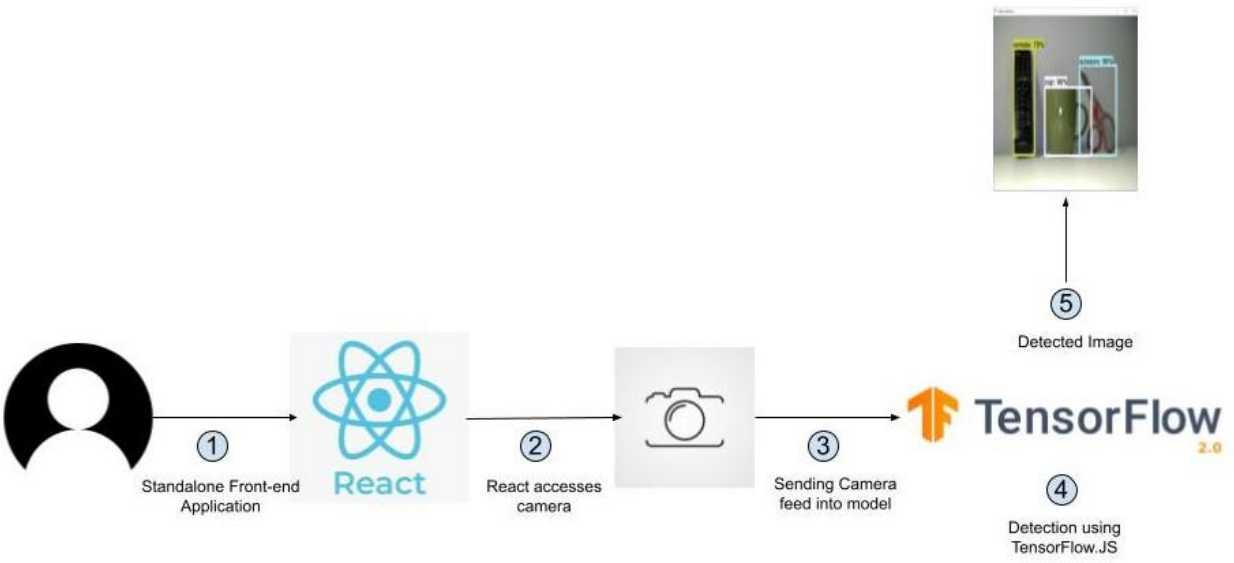



Fig 5.1: Data Flow Diagram

1) **Import modules**

   This process begins by initially importing all the necessary modules like OpenCV, TensorFlow, Node JS, React, etc.

2) **Creating a standalone Front-end application**

   After importing all the necessary modules, we create a standalone Front-end web application using React for accessing camera footage, displaying predicted information and for easier usage in any type of device.

3) **Sending a camera feed into the model**

   The live video footage is fed from the camera into the model. The model then tries to detect multiple objects in the frame based on its confidence value.

4) **Make Detection using TensorFlow.JS Model**

   A machine learning model is trained and converted using Tensorflow.js. Tensorflow.js model directly detects the camera feed and detects different objects based on the adjusted weights of the model. Tensorflow.js is an ML library for JavaScript. It helps to deploy machine learning models directly into node.js or a web browser.

5) **Displaying the detected objects**

After the detection of the objects, the classified object is differentiated by the colored square and the name of the object on the left corner of the square.

## 5.1.2 Architecture:

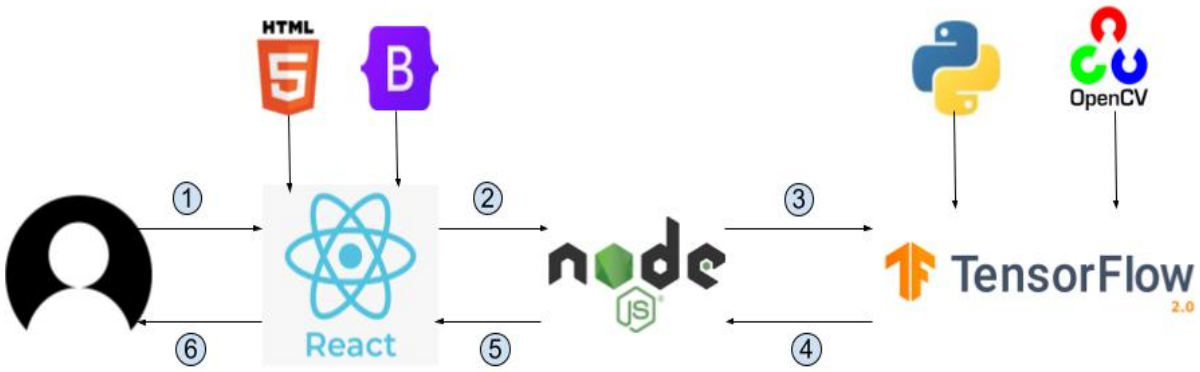



Fig 5.2: Architecture Diagram

1) **User Interaction with Frontend:**

React is a JavaScript library that will be used for building user-friendly interfaces. React is designed to solve the issues of partial content updates for web pages, which can be found in the development of one-page applications. Compared to other frontend frameworks, the React code is easier to maintain and is flexible due to its modular structure. This flexibility, in turn, saves a huge amount of time.

2) **Requesting Connection from Frontend to Backend:**

Next, we will connect the React frontend to a backend. For that, we will use Node JS which is primarily used for non-blocking, event-driven servers, due to its single-threaded nature. It's used for traditional websites and back-end API services but was designed with real-time, push-based architectures in mind. To make this connection we just have to fetch data from the port that the backend server is located on.

3) **Connecting Backend with TensorFlow.js Model:**

Here we will deploy a Tensorflow.js model using NodeJS. Tensorflow.js is an ML library for JavaScript. It helps to deploy machine learning models directly into node.js or a web browser.

4) **Fetching Machine Learning Model TensorFlow.js:**

Now we will use TensorFlow to train a custom object-detection model in Python, then put it into production, and run real-time inferences in the browser through TensorFlow.js.

5) **Processing the user request in accordance with TensorFlow.js Model:**

TensorFlow.js provides functionality for saving and loading models that have been created with the Layers API or converted from existing TensorFlow models. When we run it on Node.js we also have direct access to the filesystem and can save models there.

6) **Fetching the required output:**

The video feed is processed by the Tensorflow.js model which detects different objects and creates colored boxes for the different objects along with the label.

# 6. SYSTEM IMPLEMENTATION

This section covers the project's System Implementation phase. This section also includes a detailed description of each step. Data pre-processing, model initialization, and the training procedure are the steps that have been covered in this section. This section also includes information on the web interface implementation process.

## 6.1 Data Preprocessing

Firstly the model is trained on a known image database. For this, we have used the coco dataset which contains images in 80 different categories having a total of more than 330k images including 200k labeled images. For our use, we pick a very popular deep learning model which is used widely and will alter its layers as per our requirement using transfer learning.

There are various steps in data preprocessing which is required to perform before the model can be trained using the images

1. The size of the image is decided for input to the neural network and the images which will be used for training are adjusted accordingly.
2. Since we are applying alterations and adding layers to a pre-trained deep learning model which already takes a 3D array(of the image) as input we don't need to make any changes to the image before sending it to the neural network.
3. Then before passing the image to the neural network for training we image augmentation to tweak the images slightly by rescaling, rotating, flipping, right shifting, left shifting, etc for improving the quality of the data the model is trained on. This way the ability of the model to detect objects at different angles and even if part of the object is present can be detected.

## 6.2 Implementation of MobileNet50

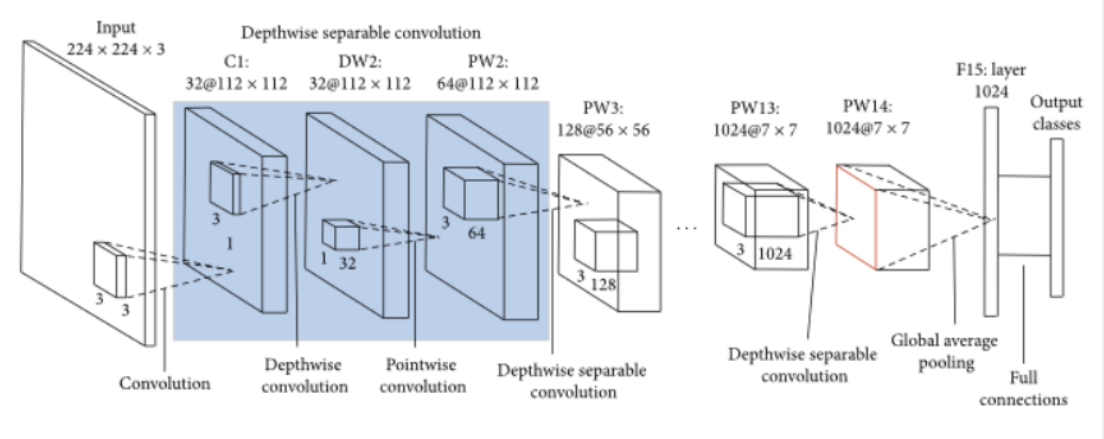



Fig 6.1: MobileNet50 structure

The MobileNet model is designed to be used in mobile applications, and it is TensorFlow's first mobile computer vision model. MobileNet uses depthwise separable convolutions. It significantly reduces the number of parameters when compared to the network with regular convolutions with the same depth in the nets. This results in lightweight deep neural networks. A depthwise separable convolution is made from two operations.

- Depthwise convolution.
- Pointwise convolution.

MobileNet is a class of CNN that was open-sourced by Google, and therefore, this gives us an excellent starting point for training our classifiers that are insanely small and insanely fast.

| Type / Stride | Filter Shape | Input Size |
|---|---|---|
| Conv / s2 | $3 \times 3 \times 3 \times 32$ | $224 \times 224 \times 3$ |
| Conv dw / s1 | $3 \times 3 \times 32$ dw | $112 \times 112 \times 32$ |
| Conv / s1 | $1 \times 1 \times 32 \times 64$ | $112 \times 112 \times 32$ |
| Conv dw / s2 | $3 \times 3 \times 64$ dw | $112 \times 112 \times 64$ |
| Conv / s1 | $1 \times 1 \times 64 \times 128$ | $56 \times 56 \times 64$ |
| Conv dw / s1 | $3 \times 3 \times 128$ dw | $56 \times 56 \times 128$ |
| Conv / s1 | $1 \times 1 \times 128 \times 128$ | $56 \times 56 \times 128$ |
| Conv dw / s2 | $3 \times 3 \times 128$ dw | $56 \times 56 \times 128$ |
| Conv / s1 | $1 \times 1 \times 128 \times 256$ | $28 \times 28 \times 128$ |
| Conv dw / s1 | $3 \times 3 \times 256$ dw | $28 \times 28 \times 256$ |
| Conv / s1 | $1 \times 1 \times 256 \times 256$ | $28 \times 28 \times 256$ |
| Conv dw / s2 | $3 \times 3 \times 256$ dw | $28 \times 28 \times 256$ |
| Conv / s1 | $1 \times 1 \times 256 \times 512$ | $14 \times 14 \times 256$ |
| $5\times$ Conv dw / s1 | $3 \times 3 \times 512$ dw | $14 \times 14 \times 512$ |
| $5\times$ Conv / s1 | $1 \times 1 \times 512 \times 512$ | $14 \times 14 \times 512$ |
| Conv dw / s2 | $3 \times 3 \times 512$ dw | $14 \times 14 \times 512$ |
| Conv / s1 | $1 \times 1 \times 512 \times 1024$ | $7 \times 7 \times 512$ |
| Conv dw / s2 | $3 \times 3 \times 1024$ dw | $7 \times 7 \times 1024$ |
| Conv / s1 | $1 \times 1 \times 1024 \times 1024$ | $7 \times 7 \times 1024$ |
| Avg Pool / s1 | Pool $7 \times 7$ | $7 \times 7 \times 1024$ |
| FC / s1 | $1024 \times 1000$ | $1 \times 1 \times 1024$ |
| Softmax / s1 | Classifier | $1 \times 1 \times 1000$ |

Fig 6.2: Architecture of MobileNet

### 6.2.1 Depthwise Separable Convolution

This convolution originated from the idea that a filter's depth and spatial dimension can be separated- thus, the name is separable.

1. Depthwise convolution is the channel-wise DK×DK spatial convolution. Suppose in the figure above, we have five channels; then, we will have 5 DK×DK spatial convolutions.
2. Pointwise convolution is the 1×1 convolution to change the dimension.

3. Depthwise convolution

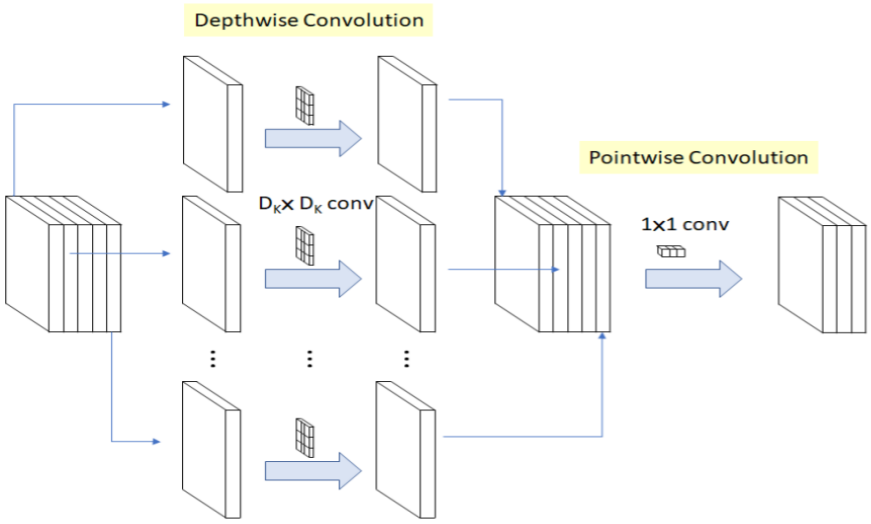



Fig 6.3: Depthwise Separable Convolution Diagram

## 6.3 Web Interface Implementation

An interface was developed using Javascript, CSS, and HTML. This combination was hosted on a React Application which deploys our interface on a port and starts listening and monitoring for any requests. The interface has the following features:

• Interface is hosted on localhost port: 3000

• Prompts the user for video input.

• Video input can be provided as soon as the user allows it.

• Proper interfaces have been provided for both these methods.

• On clicking the launch Button, the interface opens the camera for taking live footage

• The live video footage gets classified and labeled based on the object present in the current frame.

• All the necessary computations are performed by the tensorflow.js model

• Final classification result is displayed in the feed and can be seen by a square box with different colors.

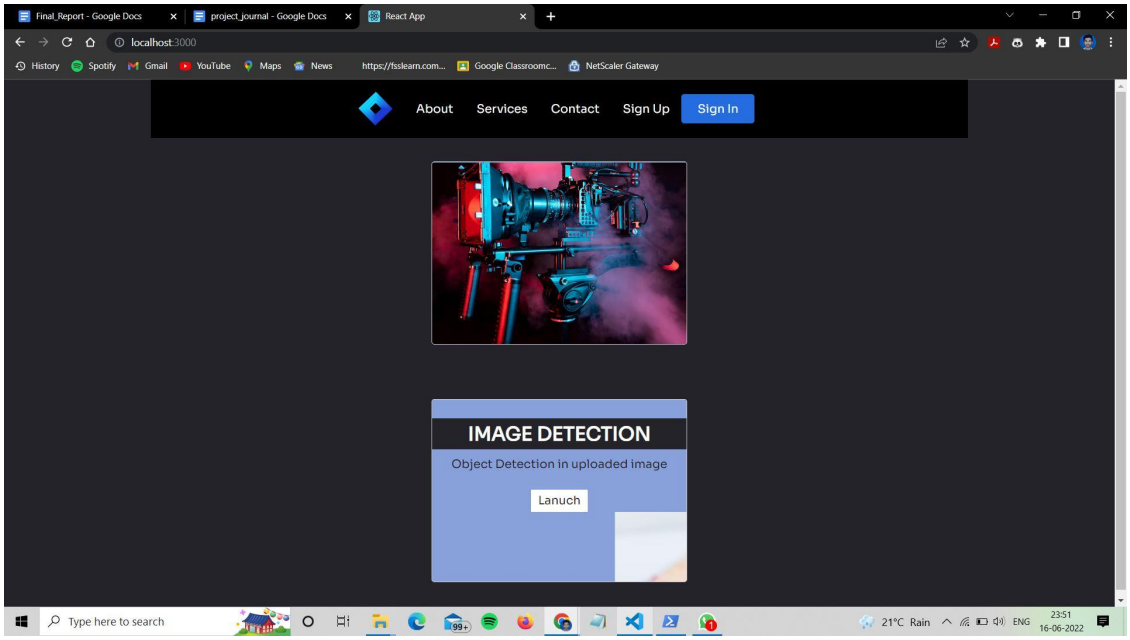


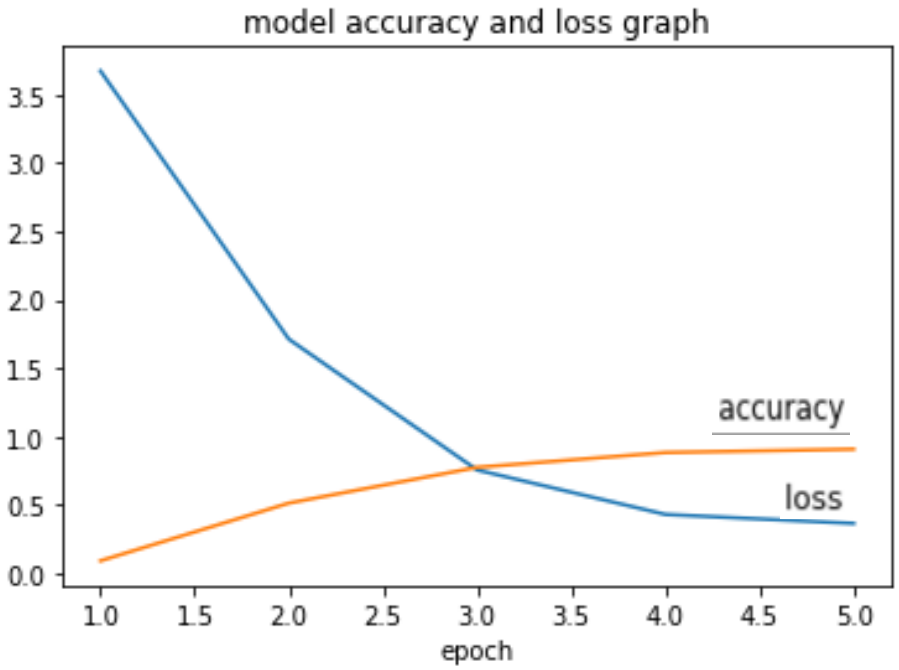

Fig 6.4: Landing Page

# 7. SYSTEM TESTING AND RESULT ANALYSIS

Unit testing focuses verification effort on the smallest unit of software design: the software component or module.We developed and trained all the architectures based on the steps mentioned in the System implementation section. We plotted graphs at the final step of each architecture that we implemented. Various test cases were developed to check the functionality of our web page and all the test cases were provided with appropriate output.

## 7.1 Model Implementation :

We have obtained the following accuracy and loss chart:




We were finally able to reduce the loss till 0.36 and achieve training accuracy of 0.90 on running it for 5 epochs.

Later on testing it with random data we were able to achieve an accuracy of 88%.

## 7.2 Interface Testing

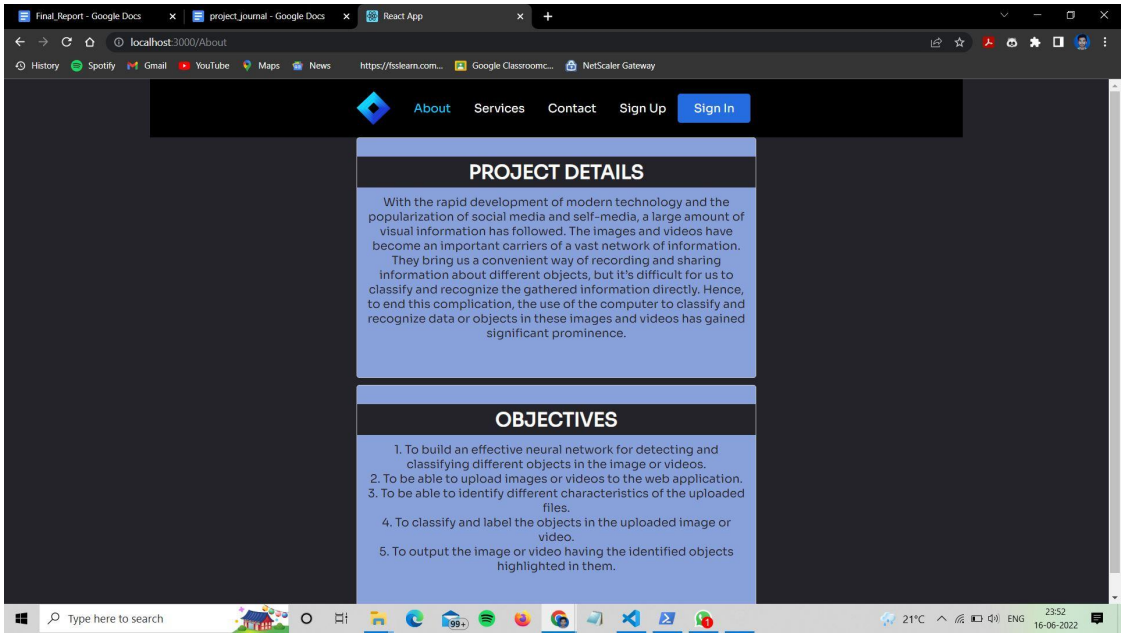



Fig 7.3: Interface Screen (About)

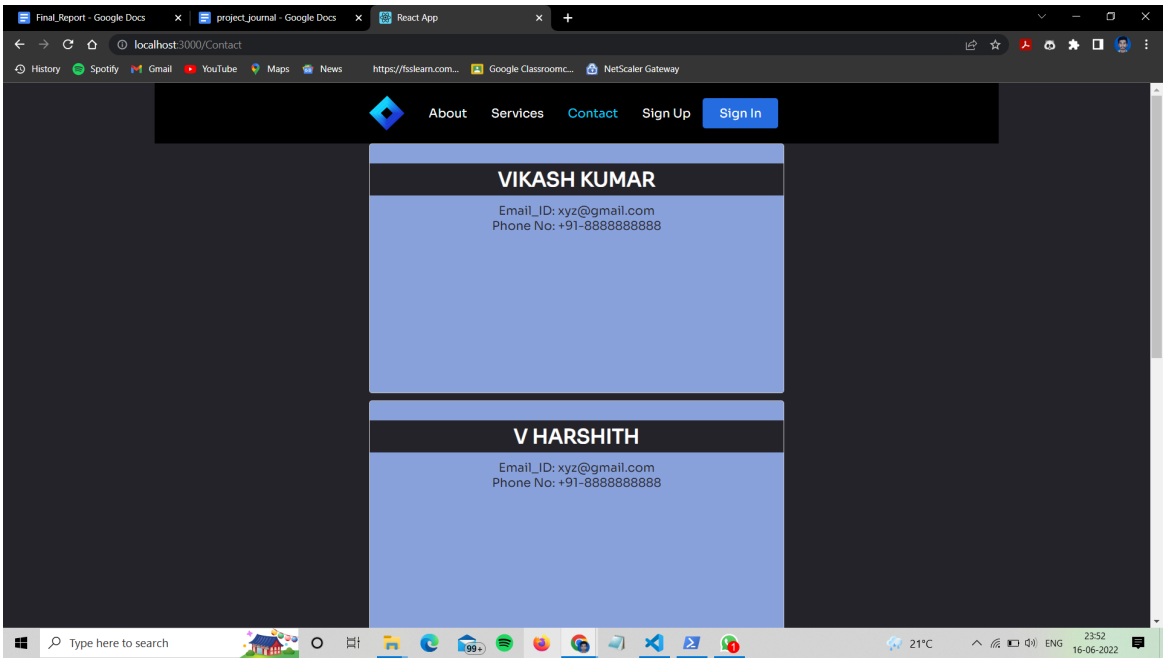



Fig 7.4: Interface Screen (Contact)

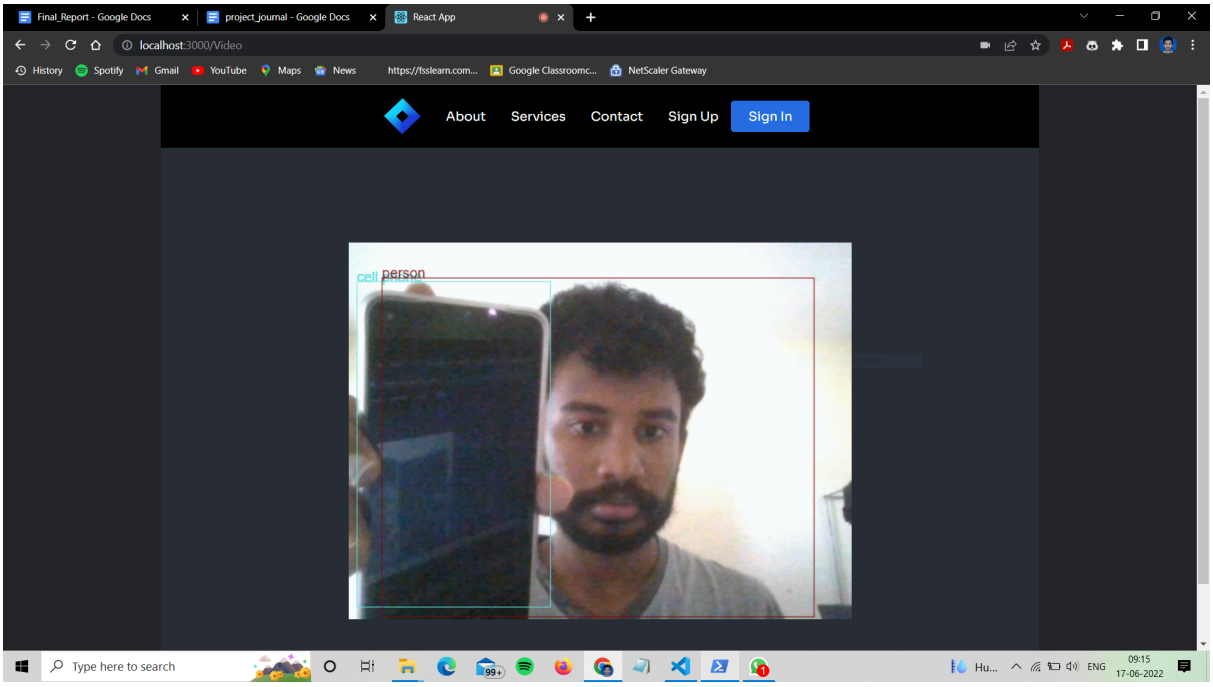



Fig 7.5: Interface Test on a Real Video

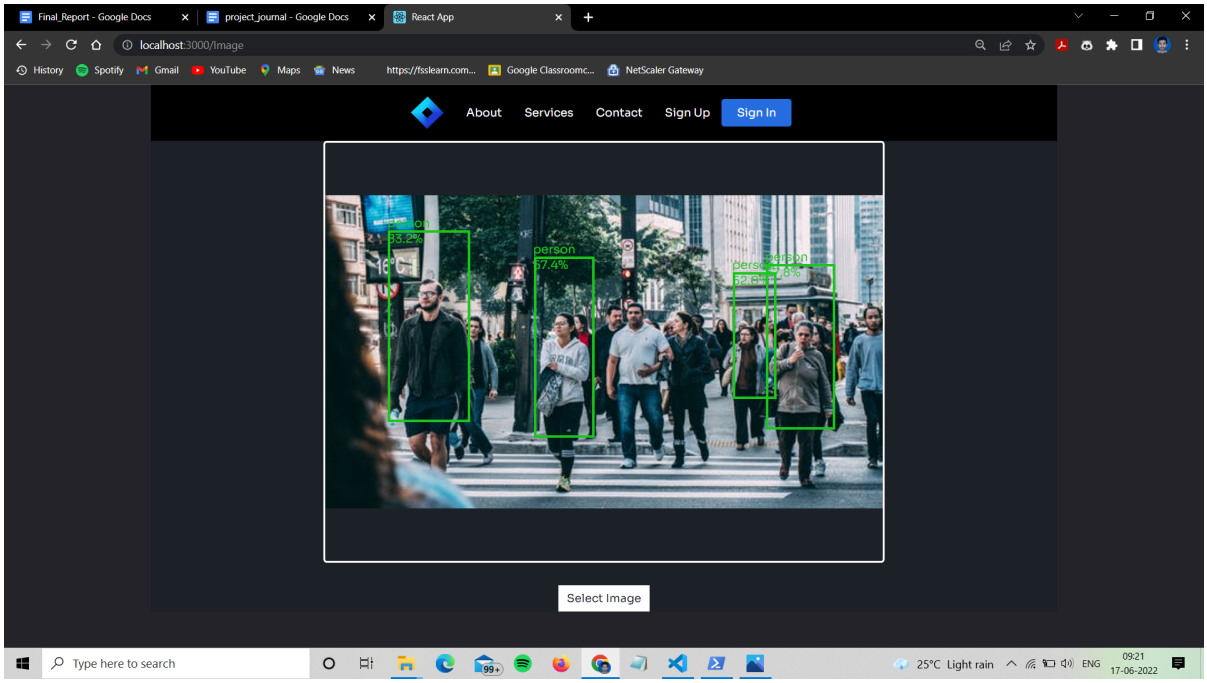



Fig 7.6: Interface Test on the Image

# 8. CONCLUSION AND FUTURE WORK

In this work, we improve the ability of the convolutional neural network to classify and recognize two-dimensional images and speed up the convergence of the algorithm.

To summarize the project in a few short points:

i. MobileNet50 was studied and implemented.

ii. Observing the limitations of neural architectures was explored.

Video object detection is both an important research topic and a complex problem in practical applications. As video object detection is a popular and promising field in the field of machine learning, multiple object detection algorithms have been established, and the demands for video data processing have gradually increased. But at present, the computational loads of the existing algorithms and the ability to achieve real-time performance levels are the biggest constraint as they restrict the applications of these algorithms.

Also detecting object motion in a real-world situation is one of the challenges we face since object detection here is a continuous process. During the detection defocusing is caused because the imaging device fails to always focus on the target accurately. In defocus conditions, target objects will appear unclear and blurry. Therefore, the features extracted by the network are also less clear, making such objects difficult to detect. As a future work potential improvements could be included as detection algorithms become mature, thereby increasing the accuracy and performance of video object detection.

# 9. REFERENCES

[1] Lijuan Liu, Yanping Wang, and Wanle Chi, “Image Recognition Technology Based on Machine Learning”.

[2] Xiao H, Rasul K, and Vollgraf R. Fashion-MNIST: “A Novel Image Dataset for Benchmarking Machine Learning Algorithms”.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep Residual Learning for Image Recognition”.

[4] S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun., “Object detection networks on convolutional feature maps”.

[5] Licheng Jiao, Ruohan Zhang, and Fang Liu, “New Generation Deep Learning For Video Object Detection”.

[6] L. Jiao et al., “A survey of deep learning-based object detection”.

[7] Joseph Redmon, Santosh Divvala, and Ross Girshick, “You Only Look Once: Unified, Real-Time Object Detection”.

[8] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes challenge: A retrospective”.