MAJOR-2 PROJECT
End Semester Report
Solution for mask and temperature detection for large gatherings.

Submitted By:

Name	Roll No	Branch
Aanchal Sharma	R110218002	CSE-CCVT
Arnav Sharma	R110218032	CSE-CCVT
Aman Sharma	R110218018	CSE-CCVT
Gaurvendra Singh	R110218056	CSE-CCVT

Under the guidance of:

Dr. Anurag Jain
Systemics Cluster
School of Computer Science

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES

Bidholi Campus, Energy Acres
Dehradun-248007

CANDIDATE'S DECLARATION

We hereby certify that the project work entitled Solution for mask and temperature detection for large gatherings is partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering with Specialization in Cloud Computing and Virtual Technology and submitted to the Department of Virtualization at School of Computer Science, University of Petroleum and Energy Studies, Dehradun, is an authentic record of my/our work carried out during a period from Jan, 2022 to May, 2022 under the supervision of Anurag Jain,

The matter presented in this project has not been submitted by us for the award of any other degree of this or any other University.

Arnav Sharma	Aanchal Sharma	Aman Sharma	Gaurvendra Singh
R110218032	R110218002	R110218018	R110218056

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: 05/05/2022

Dr. Neelu J. Ahuja
Head of Department
Systemics Cluster
School of Computer Science

Dr. Anurag Jain
Associate Professor
Systemics Cluster
School of Computer Science

Acknowledgement

We wish to express our deep gratitude to our guide Dr. Anurag Jain Sir, for all the advice, encouragement and constant support he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions.
We sincerely thank our Head of Department Dr. Neelu J. Ahuja , for great support in doing our project at SoCS.

We would like to thank all our friends for their help and constructive criticism during our project work. Finally, we have no words to express our sincere gratitude to our parents who have shown us this world and for every support they have given us.

Arnav Sharma	Aanchal Sharma	Aman Sharma	Gaurvendra Singh
R110218032	R110218002	R110218018	R110218056

Table Of Content

Topic	Page No.
1. Project Title	6
2. Abstract	6
3. Introduction	7
4. Literature Review	15
5. Problem Statement	22
6. Objectives	22
7. Flow Chart	23
8. System Requirements	24
9. Methodology	25
10. Results and Discussion	33
11. Pert Chart	36
12. Conclusion	37
13. References	39

Table Of Figures

Topic	Page No.
1. Types of Machine Learning	8
2. Deep Learning Model	10
3. Transfer Learning	11
4. Xception Model	13
5. ResNet Model	14
6. Flow Chart	23
7. Test Result Images	33
8. Pert Chart	36

End Sem Report (2022)

Major 2

Project Title:

Solution for mask and temperature detection for large gatherings

ABSTRACT

Since the infectious coronavirus disease (COVID-19) was first reported in Wuhan, it has become a public health problem in China and even around the world. This pandemic is having devastating effects on societies and economies around the world. The increase in the number of COVID-19 tests gives more information about the epidemic spread, which may lead to the possibility of surrounding it to prevent further infections. However, wearing a face mask that prevents the transmission of droplets in the air and maintaining an appropriate physical distance between people, and reducing close contact with each other can still be beneficial in combating this pandemic. Therefore, this Project focuses on implementing a Face Mask and Social Distancing Detection model as an embedded vision system. The pretrained models such as the MobileNet, ResNet Classifier, and VGG are used in our context. People violating social distancing or not wearing masks were detected. After implementing and deploying the models, the selected one achieved a confidence score of 100%. This project also provides a comparative study of different face detection and face mask classification models. The system performance is evaluated in terms of precision, recall, F1-score, support, sensitivity, specificity, and accuracy that demonstrate the practical applicability. The system performs with F1-score of 99%, sensitivity of 99%, specificity of 99%, and an accuracy of 100%. Hence, this solution tracks the people with or without masks in a real-time scenario and ensures social distancing by generating an alarm if there is a violation in the scene or in public places. This can be used with the existing embedded camera infrastructure to enable these analytics which can be applied to various verticals, as well as in an office building or at airport terminals/gates.

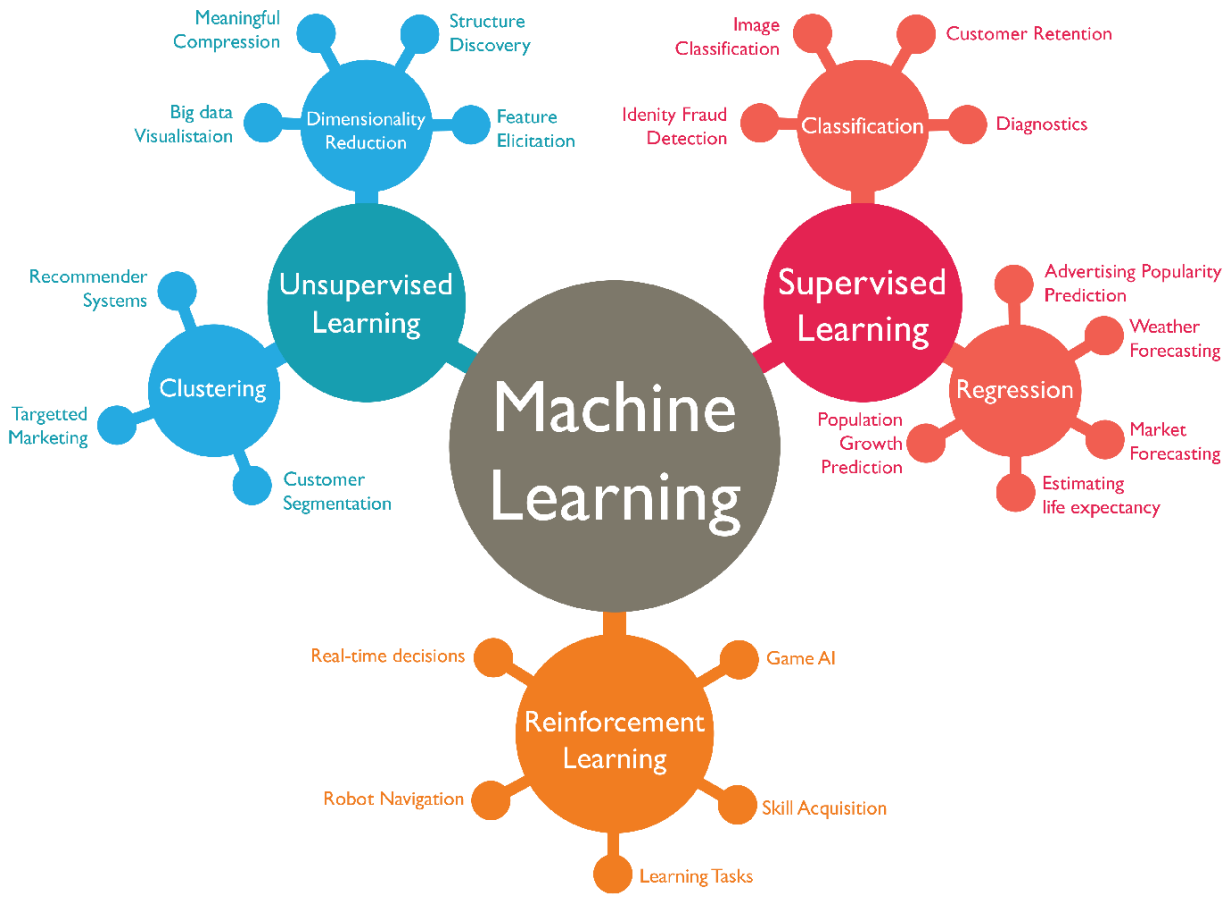
INTRODUCTION

Currently, we have 476 million confirmed cases and near 6 million deaths worldwide due to the coronavirus pandemic. While some factors can be eluded as being not being under our control as individuals, we cannot say that face masks do not have a big role to play in controlling this pandemic. Upon wearing the face mask, we as the source of infection reduce the number of droplets ejected by 99 percent according to confirmed studies, and if we reduce the number of people getting infected, it also reduces the effective reproduction rate, and hence having an impact of exponential margins. Nearly half of the infected people do not show symptoms as per recent studies, which can take up to a period of 14 days to appear in an infected individual. Hence, it is really necessary for wearing masks by people in public places, and should really be made mandatory rather than being based on individual decisions, as a significant portion of people with infection lack coronavirus symptoms.

In this project, we have developed a Realtime the mask detection model can be said to be a combination of classification and face detection model. For the purpose of classification, we use transfer learning with an Xception model trained on the ImageNet dataset with a modified final fully connected layer. While using the face detection model, several different approaches were tried upon based on existing literature, and the one which worked the best was a RetinaNet Face pre-trained model which gave the highest measures of recall while experimenting on different use-cases and testing images of people in a crowded setting. The models and implementation details for them have been discussed in an objective manner as part of this section, and while providing an insight on the approach used (and why it was chosen in the first place), we delve into our final mask detection model which was built using a combination of the classification and face detection models as were briefly described above.

Machine Learning:

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.



1) Types of Machine Learning

Types:

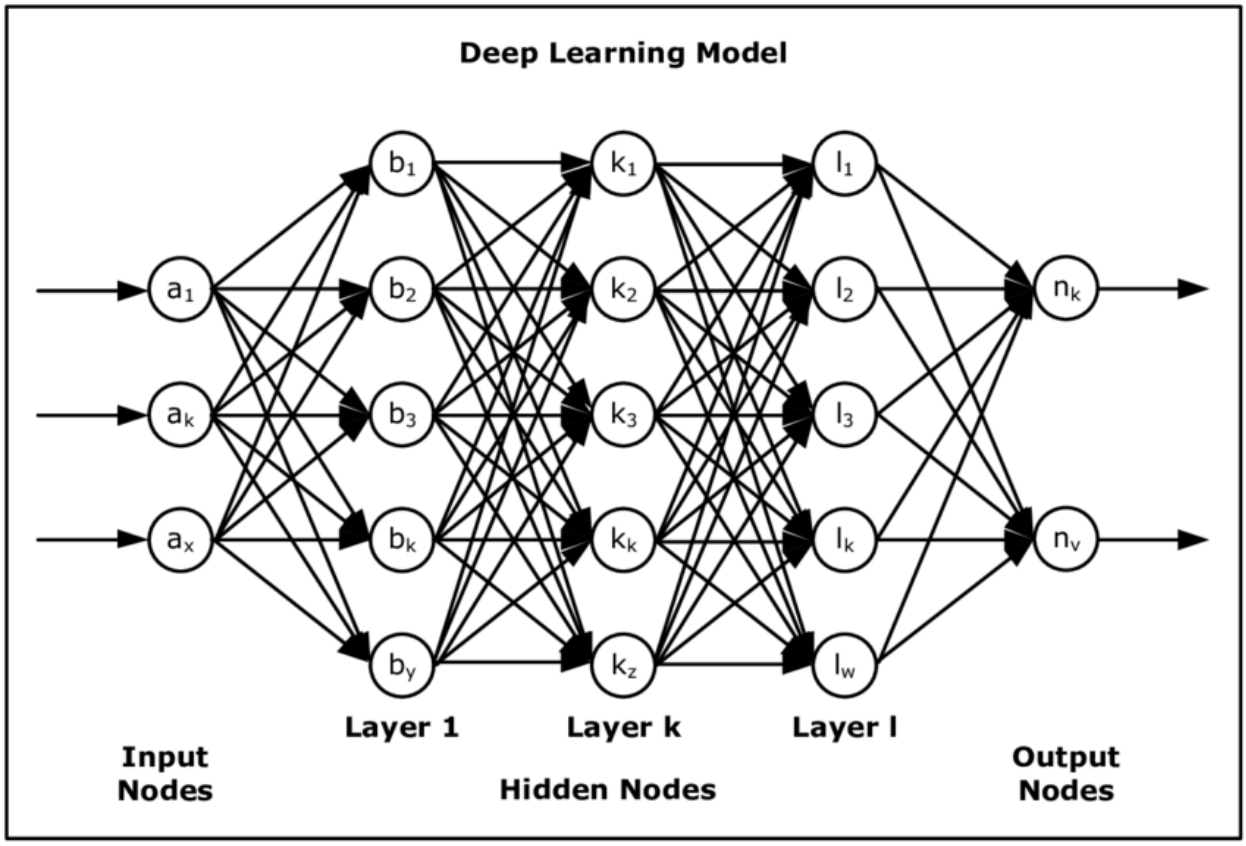
Supervised learning: In this type of machine learning, data scientists provide algorithms with labeled training data and define the alternatives they want the algorithm to test for relevance. Both input and output algorithm are specified.

Unsupervised learning: This type of machine learning involves algorithms that are trained in non-labeled data. The algorithm checks on data sets looking for any logical connection. Data trained by algorithms and predictions or outgoing recommendations are predetermined.

Semi-supervised learning: This method of machine learning involves a combination of the two previous types. Data scientists may supply an algorithm with a training data label, but the model is free to test the data itself and improve its understanding of the data set.

Reinforcement reading: Data scientists often use reinforcement learning to teach the machine to complete a multi-step process where there are clearly defined rules. Data scientists devise an algorithm to complete the task and give it a good or bad idea as they explore how to complete the task. But for the most part, the algorithm determines for itself which steps to take.

Deep learning:

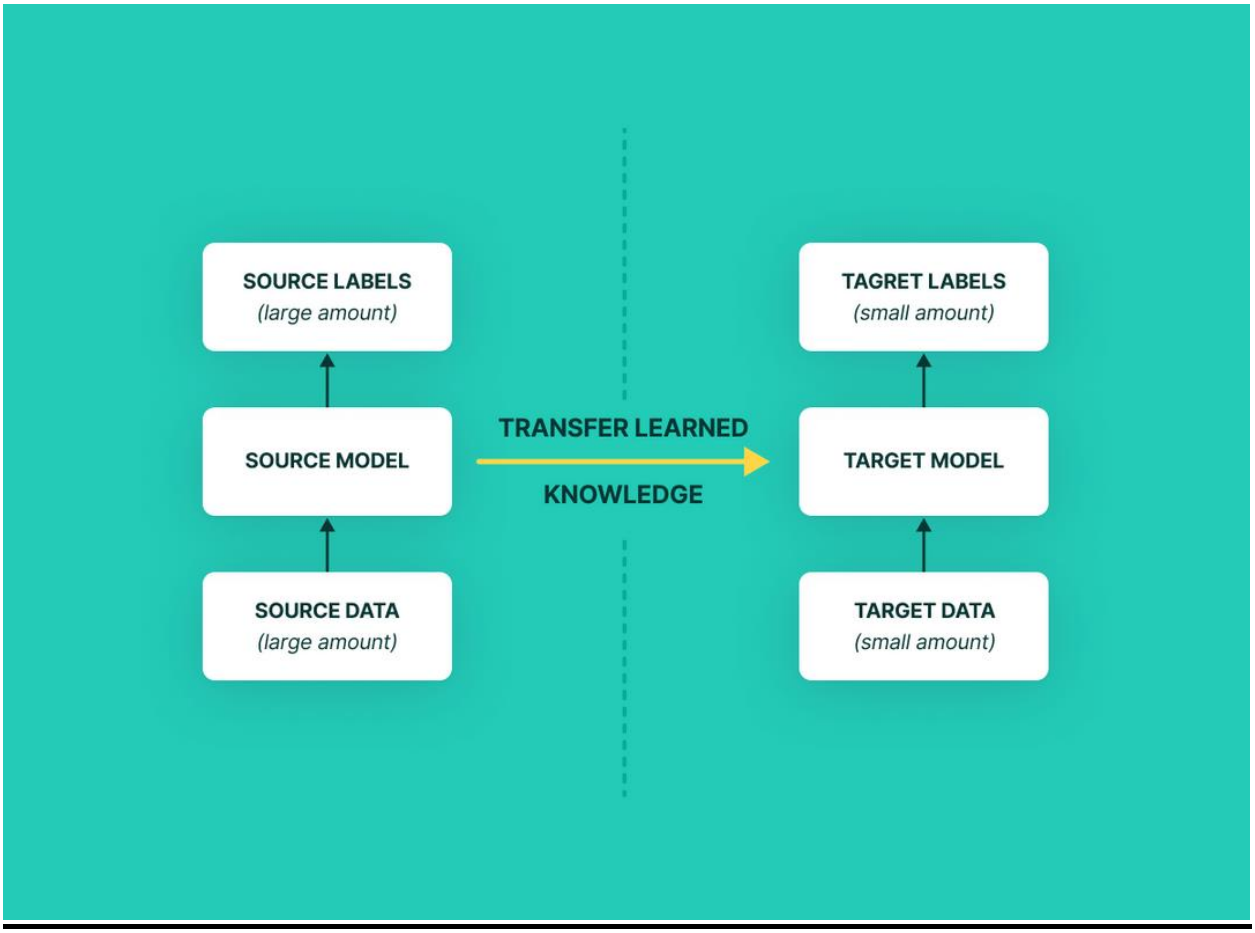


2) Deep Learning Model

Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behaviour of the human brain—albeit far from matching its ability—allowing it to "learn" from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.

Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention. Deep learning technology lies behind everyday products and services (such as digital assistants, voice-enabled TV remotes, and credit card fraud detection) as well as emerging technologies (such as self-driving cars).

Transfer Learning:



3) Transfer Learning

Transfer Learning is a machine learning method where we reuse a pre-trained model as the starting point for a model on a new task.

To put it simply—a model trained on one task is repurposed on a second, related task as an optimization that allows rapid progress when modelling the second task.

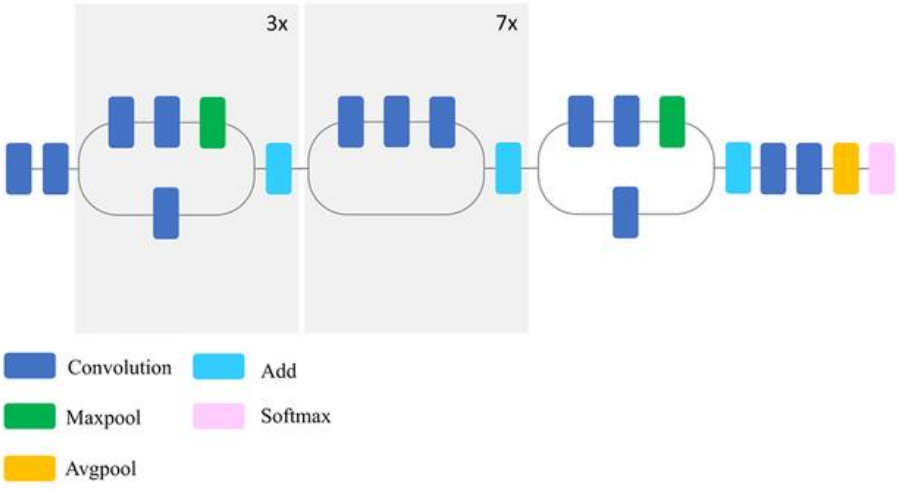
By applying transfer learning to a new task, one can achieve significantly higher performance than training with only a small amount of data.

Transfer learning is so common that it is rare to train a model for an image or natural language processing-related tasks from scratch.

Instead, researchers and data scientists prefer to start from a pre-trained model that already knows how to classify objects and has learned general features like edges, shapes in images.

ImageNet, AlexNet, and Inception are typical examples of models that have the basis of Transfer learning.

Xception:



4) Xception

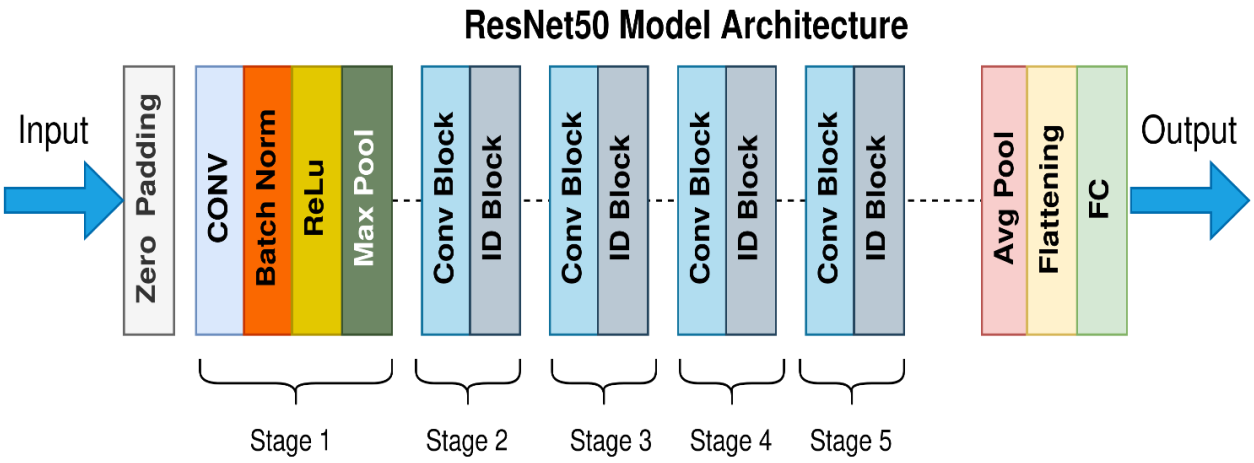
Xception is a deep convolutional neural network architecture that involves Depthwise Separable Convolutions. This network was introduced Francois Chollet who works at Google, Inc.
Xception is also known as “extreme” version of an Inception module. Hence, let us look at the Inception module before delving into Xception.

Inception Network

An inception network is a deep neural network (DNN) with a design that consists of repeating modules referred to as inception modules.

The name Inceptions probably sounds familiar to some readers, especially if you are a fan of the actor Leonardo DiCaprio or movie director, Christopher Nolan. Inception (directed by Christopher Nolan) is a movie released in 2010, and the concepts of embedded dream state were the central premise of the film. This is where the name of the model was taken from.

ResNet:



5) ResNet

ResNet, short for Residual Networks is a classic neural network used as a backbone for many computer vision tasks. This model was the winner of ImageNet challenge in 2015. The fundamental breakthrough with ResNet was it allowed us to train extremely deep neural networks with 150+layers successfully. Prior to ResNet training very deep neural networks was difficult due to the problem of vanishing gradients.

AlexNet, the winner of ImageNet 2012 and the model that apparently kick started the focus on deep learning had only 8 convolutional layers, the VGG network had 19 and Inception or GoogleNet had 22 layers and ResNet 152 had 152 layers. In this blog we will code a ResNet-50 that is a smaller version of ResNet 152 and frequently used as a starting point for transfer learning.

Literature Review:

1. In current times, after the rapid expansion and spread of the COVID-19 outbreak globally, people have experienced severe disruption to their daily lives. One idea to manage the outbreak is to enforce people wear a face mask in public places. Therefore, automated and efficient face detection methods are essential for such enforcement. In this paper, a face mask detection model for static and real time videos has been presented which classifies the images as "with mask" and "without mask". The model is trained and evaluated using the Kaggle data-set. The gathered data-set comprises approximately about 4,000 pictures and attained a performance accuracy rate of 98%. The proposed model is computationally efficient and precise as compared to DenseNet-121, MobileNet-V2, VGG-19, and Inception-V3. This work can be utilized as a digitized scanning tool in schools, hospitals, banks, and airports, and many other public or commercial locations.

2. In the past years a lot of effort has been made in the field of face detection. The human face contains important features that can be used by vision-based automated systems in order to identify and recognize individuals. Face location, the primary step of the vision-based automated systems, finds the face area in the input image. An accurate location of the face is still a challenging task. Viola-Jones framework has been widely used by researchers in order to detect the location of faces and objects in a given image. Face detection classifiers are shared by public communities, such as OpenCV. An evaluation of these classifiers will help researchers to choose the best classifier for their particular need. This work focuses of the evaluation of face detection classifiers minding facial landmarks.

3. We present an interpretation of Inception modules in convolutional neural networks as being an intermediate step in-between regular convolution and the depthwise separable convolution operation (a depthwise convolution followed by a pointwise convolution). In this light, a depthwise separable convolution can be understood as an Inception module with a maximally large number of towers. This observation leads us to propose a novel deep convolutional neural network architecture inspired by Inception, where

Inception modules have been replaced with depthwise separable convolutions. We show that this architecture, dubbed Xception, slightly outperforms Inception V3 on the ImageNet dataset (which Inception V3 was designed for), and significantly outperforms Inception V3 on a larger image classification dataset comprising 350 million images and 17,000 classes. Since the Xception architecture has the same number of parameters as Inception V3, the performance gains are not due to increased capacity but rather to a more efficient use of model parameters.

4. Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we evaluate residual nets with a depth of up to 152 layers - 8× deeper than VGG nets but still having lower complexity. An ensemble of these residual nets achieves 3.57% error on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task. We also present analysis on CIFAR-10 with 100 and 1000 layers. The depth of representations is of central importance for many visual recognition tasks. Solely due to our extremely deep representations, we obtain a 28% relative improvement on the COCO object detection dataset. Deep residual nets are foundations of our submissions to ILSVRC & COCO 2015 competitions1, where we also won the 1st places on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation.

5. Though tremendous strides have been made in uncontrolled face detection, accurate and efficient face localisation in the wild remains an open challenge. This paper presents a robust single-stage face detector, named RetinaFace, which performs pixel-wise face localisation on various scales of faces by taking advantages of joint extra-supervised and self-supervised multi-task learning. Specifically, We make contributions in the following five aspects: We manually annotate five facial landmarks on the WIDER FACE dataset

and observe significant improvement in hard face detection with the assistance of this extra supervision signal. We further add a self-supervised mesh decoder branch for predicting a pixel-wise 3D shape face information in parallel with the existing supervised branches. On the WIDER FACE hard test set, RetinaFace outperforms the state-of-the-art average precision (AP) by \$1.1\%\$ (achieving AP equal to {\bf \$91.4\%\$}). On the IJB-C test set, RetinaFace enables state of the art methods (ArcFace) to improve their results in face verification (TAR=\$89.59\%\$ for FAR=1e-6). By employing light-weight backbone networks, RetinaFace can run real-time on a single CPU core for a VGA-resolution image.

6. The term Deep Learning or Deep Neural Network refers to Artificial Neural Networks (ANN) with multi layers. Over the last few decades, it has been considered to be one of the most powerful tools, and has become very popular in the literature as it is able to handle a huge amount of data. The interest in having deeper hidden layers has recently begun to surpass classical methods performance in different fields; especially in pattern recognition. One of the most popular deep neural networks is the Convolutional Neural Network (CNN). It take this name from mathematical linear operation between matrixes called convolution. CNN have multiple layers; including convolutional layer, non-linearity layer, pooling layer and fully-connected layer. The convolutional and fully-connected layers have parameters but pooling and non-linearity layers don't have parameters. The CNN has an excellent performance in machine learning problems. Specially the applications that deal with image data, such as largest image classification data set (Image Net), computer vision, and in natural language processing (NLP) and the results achieved were very amazing. In this paper we will explain and define all the elements and important issues related to CNN, and how these elements work. In addition, we will also state the parameters that effect CNN efficiency. This paper assumes that the readers have adequate knowledge about both machine learning and artificial neural network.

7. We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error

rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called "dropout" that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

8. In this paper we describe a new mobile architecture, MobileNetV2, that improves the state-of-the-art performance of mobile models on multiple tasks and benchmarks as well as across a spectrum of different model sizes. We also describe efficient ways of applying these mobile models to object detection in a novel framework we call SSDLite. Additionally, we demonstrate how to build mobile semantic segmentation models through a reduced form of DeepLabv3 which we call Mobile DeepLabv3.
   The MobileNetV2 architecture is based on an inverted residual structure where the input and output of the residual block are thin bottleneck layers opposite to traditional residual models which use expanded representations in the input an MobileNetV2 uses lightweight depthwise convolutions to filter features in the intermediate expansion layer. Additionally, we find that it is important to remove non-linearities in the narrow layers in order to maintain representational power. We demonstrate that this improves performance and provide an intuition that led to this design. Finally, our approach allows decoupling of the input/output domains from the expressiveness of the transformation, which provides a convenient framework for further analysis. We measure our performance on ImageNet classification, COCO object detection, VOC image segmentation. We evaluate the trade-offs between accuracy, and number of operations measured by multiply-adds (MAdd), as well as the number of parameters.

9. The explosion of image data on the Internet has the potential to foster more sophisticated and robust models and algorithms to index, retrieve, organize and interact with images and multimedia data. But exactly how such data can be harnessed and organized remains a critical problem. We introduce here a new database called “ImageNet”, a large-scale ontology of images built upon the backbone of the WordNet structure. ImageNet aims to populate the majority of the 80,000 synsets of WordNet with an average of 500–1000 clean and full resolution images. This will result in tens of millions of annotated images organized by the semantic hierarchy of WordNet. This paper offers a detailed analysis of ImageNet in its current state: 12 subtrees with 5247 synsets and 3.2 million images in total. We show that ImageNet is much larger in scale and diversity and much more accurate than the current image datasets. Constructing such a large-scale database is a challenging task. We describe the data collection scheme with Amazon Mechanical Turk. Lastly, we illustrate the usefulness of ImageNet through three simple applications in object recognition, image classification and automatic object clustering. We hope that the scale, accuracy, diversity and hierarchical structure of ImageNet can offer unparalleled opportunities to researchers in the computer vision community and beyond.

10. We present a fast, fully parameterizable GPU implementation of Convolutional Neural Network variants. Our feature extractors are neither carefully designed nor pre-wired, but rather learned in a supervised way. Our deep hierarchical architectures achieve the best published results on benchmarks for object classification (NORB, CIFAR10) and handwritten digit recognition (MNIST), with error rates of 2.53%, 19.51%, 0.35%, respectively. Deep nets trained by simple back-propagation perform better than more shallow ones. Learning is surprisingly rapid. NORB is completely trained within five epochs. Test error rates on MNIST drop to 2.42%, 0.97% and 0.48% after 1, 3 and 17 epochs, respectively.

11. Traditional methods of computer vision and machine learning cannot match human performance on tasks such as the recognition of handwritten digits or traffic signs. Our biologically plausible deep artificial neural network architectures can. Small (often minimal) receptive fields of convolutional

winner-take-all neurons yield large network depth, resulting in roughly as many sparsely connected neural layers as found in mammals between retina and visual cortex. Only winner neurons are trained. Several deep neural columns become experts on inputs preprocessed in different ways; their predictions are averaged. Graphics cards allow for fast training. On the very competitive MNIST handwriting benchmark, our method is the first to achieve near-human performance. On a traffic sign recognition benchmark it outperforms humans by a factor of two. We also improve the state-of-the-art on a plethora of common image classification benchmarks.

12. COVID-19 pandemic caused by novel coronavirus is continuously spreading until now all over the world. The impact of COVID-19 has been fallen on almost all sectors of development. The healthcare system is going through a crisis. Many precautionary measures have been taken to reduce the spread of this disease where wearing a mask is one of them. In this paper, we propose a system that restrict the growth of COVID-19 by finding out people who are not wearing any facial mask in a smart city network where all the public places are monitored with Closed-Circuit Television (CCTV) cameras. While a person without a mask is detected, the corresponding authority is informed through the city network. A deep learning architecture is trained on a dataset that consists of images of people with and without masks collected from various sources. The trained architecture achieved 98.7% accuracy on distinguishing people with and without a facial mask for previously unseen test data. It is hoped that our study would be a useful tool to reduce the spread of this communicable disease for many countries in the world.

13. Nowadays, automatic disease detection has become a crucial issue in medical science due to rapid population growth. An automatic disease detection framework assists doctors in the diagnosis of disease and provides exact, consistent, and fast results and reduces the death rate. Coronavirus (COVID-19) has become one of the most severe and acute diseases in recent times and has spread globally. Therefore, an automated detection system, as the fastest diagnostic option, should be implemented to impede COVID-19 from spreading. This paper aims to introduce a deep learning technique based on the combination of a convolutional neural network (CNN) and long

short-term memory (LSTM) to diagnose COVID-19 automatically from X-ray images. In this system, CNN is used for deep feature extraction and LSTM is used for detection using the extracted feature. A collection of 4575 X-ray images, including 1525 images of COVID-19, were used as a dataset in this system. The experimental results show that our proposed system achieved an accuracy of 99.4%, AUC of 99.9%, specificity of 99.2%, sensitivity of 99.3%, and F1-score of 98.9%. The system achieved desired results on the currently available dataset, which can be further improved when more COVID-19 images become available. The proposed system can help doctors to diagnose and treat COVID-19 patients easily.

14. Object detection, one of the most fundamental and challenging problems in computer vision, seeks to locate object instances from a large number of predefined categories in natural images. Deep learning techniques have emerged as a powerful strategy for learning feature representations directly from data and have led to remarkable breakthroughs in the field of generic object detection. Given this period of rapid evolution, the goal of this paper is to provide a comprehensive survey of the recent achievements in this field brought about by deep learning techniques.

15. This paper presents a proposal of an intelligent video surveillance system able to detect and identify abnormal and alarming situations by analyzing object movement. The system is designed to minimize video processing and transmission, thus allowing a large number of cameras to be deployed on the system, and therefore making it suitable for its usage as an integrated safety and security solution in Smart Cities. Alarm detection is performed on the basis of parameters of the moving objects and their trajectories, and is performed using semantic reasoning and ontologies. This means that the system employs a high-level conceptual language easy to understand for human operators, capable of raising enriched alarms with descriptions of what is happening on the image, and to automate reactions to them such as alerting the appropriate emergency services using the Smart City safety network.

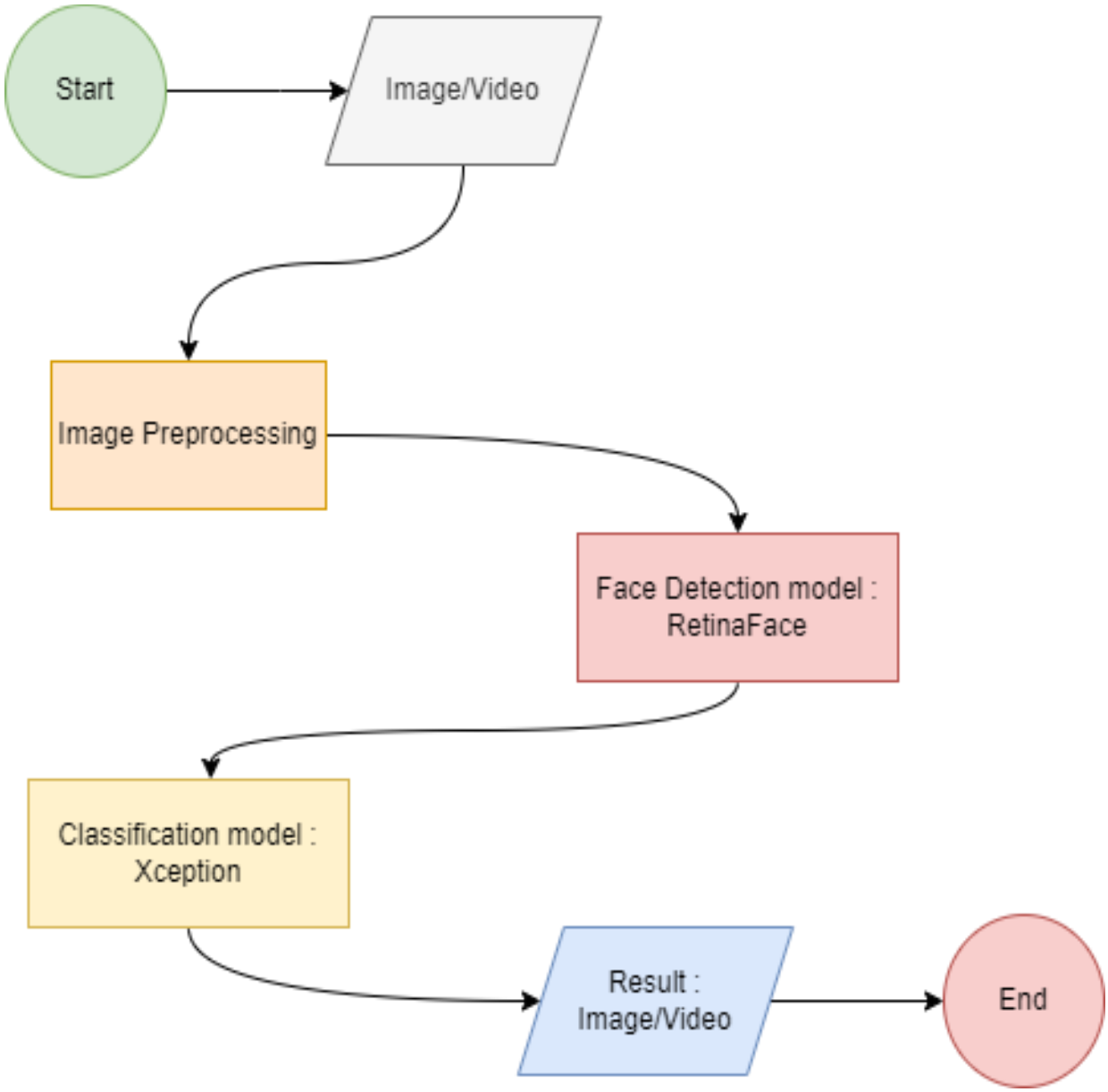
PROBLEM STATEMENT

For large crowds, manual mask and temperature is a time-consuming and inaccurate approach to use. Design and develop a solution for mask and temperature detection for large gatherings.

OBJECTIVES

- Creation of Real Time Face mask detection model.
  - Sub Objectives:
    - Creation of Front-End.
    - Creation of Deployable ML model.

Flow Chart



6) Flowchart

SYSTEM REQUIREMENT

- **Hardware Interface:**
    - RAM: 8 GiB
    - Disk Space: 1GiB (min)
    - GPU

- **Software Interface:**
    - Windows/Linux based Operating System
    - GCC Compiler.

- **Resources:**
    - TensorFlow/Keras
    - Pytorch

METHODOLOGY

As discussed earlier, the mask detection model can be said to be a combination of classification and face detection model. For the purpose of classification, we use transfer learning with an Xception model trained on the ImageNet dataset with a modified final fully connected layer. While using the face detection model, several different approaches were tried upon based on existing literature, and the one which worked the best was a RetinaNet Face pre-trained model which gave the highest measures of recall while experimenting on different use-cases and testing images of people in a crowded setting. The models and implementation details for them have been discussed in an objective manner as part of this section, and while providing an insight on the approach used (and why it was chosen in the first place), we delve into our final mask detection model which was built using a combination of the classification and face detection models as were briefly described above.

Data Collection and Pre-processing

There were issues circumventing the data collection process, primarily the unavailability of data specific to detecting face masks as detection problem. The images did not have localized bounding boxes, and did not have category labels for the masked and without mask faces of people in the image. Due to this reason, I had to use the approach for face detection and followed by the classification of faces into mask and without mask categories.

This section can be divided into two parts, with the first part focusing solely on the part of data collection for training examples, followed by the part where I discuss the collection of testing image and video examples. This was done in separate stages to test the proposed approach in a real world crowded setting of people, the use-case for which my algorithm of mask detection is proposed, which would not have been ideal in a setting wherein the training and testing data came from the same source.

1. Data Collection for Training examples

As discussed earlier, there were various issues which occurred during the collection process for training examples, due to the unavailability of datasets specific to face mask detection. A solution to the same would have been using facial landmarks to detect important indicators for faces, but this idea was not used in our implementation. Finally, it was decided that my best option would be to use a data classified into two categories of images: images of people wearing masks and the ones without mask. It is important to note that the people wearing masks on their face in an incorrect manner would be classified into the "without mask" category.

Data collection was done using a variety of sources from interfaces such as Kaggle API and Bing Search. As an additional source, the Real World Masked Face Detection (RMFD) dataset was used to facilitate training of our mask detection algorithm using the classification based approach of detecting masks followed by predicting whether the detected face contains a mask or not.
The total number of images in the training set were divided into the two categories as follows:

1. With mask: **8072 images**
2. Without mask: **8086 images**

2. Data Collection for Inference

The images and video sequence collected for the purpose of testing our model had to come from a crowded setting to provide a realistic test scenario for my proposed model for mask detection. This task was accomplished using using Google search for images and available YouTube videos for the purpose of scraping the web for realistic examples suited to inference in a real-world surveillance setting.

Reading Image

- We need to apply transformations on the input image before defining inputs for the model architecture.
- Transformations are done so as to ensure that the inputs to the network are of the right size and respective values sit in a similar range.
- Input image was resized to (128,128) for (height, width) and values in the image tensor were scaled to the range of (-1,1).
- Entire dataset was split into training and test set for cross validation. Split ratio was chose such that 90 percent of the dataset is part of the training set and 10 percent for test set.

```
def read_img(image_path):
  img = tf.io.read_file(image_path)
  img = tf.image.decode_image(img, channels=3)
  img.set_shape([None,None,3])
  img = tf.image.resize(img, [image_w, image_h])
  img  = img/127.5-1
  return img

label_map = {v:i for i, v in enumerate(classes)}

images = glob('/content/Dataset/*/*')
np.random.shuffle(images)

labels = [label_map[x.split('/')[-2]] for x in images]

(train_images, test_images, train_labels, test_labels) = train_test_split(images, labels,
        test_size=0.10, stratify=labels, random_state=42)

#reading image and label
def load_data(image_path, label):
  image = read_img(image_path)
  return image, label
```

Model for Classification: Xception

A classification problem is, using available training data with defined features and class labels, building a function which can, with high levels of certainty categorize a new unseen set of data into one of the classes. Having only a limited amount of data available for training the classifier, I was inclined to use transfer learning for the purpose of our task of classifying an input image into the categories of whether the subject is wearing a mask or not. The transfer learning technique is probably one of the most revolutionary ideas to have come out in the past few years, and can be thought of reusing a pre-trained model, which is trained on another set of input images, which in our case would be the use of Xception model trained on the ImageNet database.

Some properties of the Xception model chosen in our case are discussed below:

1. Shape invariance: similar results irrespective of the dimensions of input images (by using Global Avg pooling instaed of flatten) .
2. Removal of final fully connected layer: Prior layers to the final fully connected layer were trained using the weights of Xception trained on the ImageNet database.

```
base_model = tf.keras.applications.Xception(include_top=False,
                                            input_shape=(None, None, 3),
                                            weights='imagenet')
base_model.trainable = False
layer = tf.keras.layers.GlobalAveragePooling2D()(base_model.output)
layer = tf.keras.layers.Dense(1024, activation='relu')(layer)
layer = tf.keras.layers.Dropout(0.5)(layer)
output = tf.keras.layers.Dense(num_classes, activation='softmax')(layer)
model = tf.keras.models.Model(base_model.inputs,output)
model.summary()
```

Model fitting

Implementation details for classification architecture:

- ***Batch Size***: 32
- ***Epochs***: 2
- ***Learning rate***: 1e-4 (with decay of 1e-4 / epoch)
- ***Gradient Descent Optimizer***: Adam
- ***Loss function***: Sparse categorical cross entropy
- ***Criterion for evaluation*** (metric): F1-score

```
#We can use learning rate scheduler here like Cyclical Learning Rate(available in Tf Addons)
model.compile(optimizer=tf.keras.optimizers.Adam(1e-4, decay=1e-4 / epoch),
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

callbacks = [
    tf.keras.callbacks.ModelCheckpoint('mask_classification_model.h5',
                                       save_best_only=True,
                                       save_weights_only=True,
                                       monitor='loss')]

model.fit(train_dataset,
          batch_size=batch_size,
          epochs=epoch,steps_per_epoch=step_per_epoch,
          callbacks=callbacks)
```

Model for Face Detection: RetinaNet Face

Face Detection is the technique of identifying human faces in digital images. While inherently a backbone to other applications, detecting a face is in fact impacted a lot in cluttered scenes, and examples of the same kind of problem can be thought of in a crowded setting, the use-case for which our mask detection algorithm is being built. For this reason, having tried various techniques from classical Computer Vision domain to using deep learning techniques, I needed to prioritize the mAP metric of detected faces in a crowded setting. Having said that, I tried techniques for face detection such as Haar cascading and MT-CNN which did not achieve a high recall. Finally, I sided with a pre-trained RetinaNet Face model, using focal loss which is able to handle the foreground-background class imbalance (an issue with one stage detectors which makes performance of single shot detectors inferior to two stage detectors for object detection) in the detected classes pretty well.

Improving the precision and recall

I was able to improve the precision and recall of my model by a high margin in the following ways:

1. Resizing the cropped face before providing as input to the classification model.
2. For cases wherein the dimensions for height and width of the face crops fall below a threshold, increasing the dimensions of the crop by some proportion.

```
def visualize_detections(image, boxes):

  figsize=(7, 7)
  linewidth=1

  image = np.array(image, dtype=np.uint8)

  plt.figure(figsize=figsize)
  plt.axis('off')
  plt.imshow(image)

  ax = plt.gca()

  for box in boxes:
    x, y, w, h = box

    face_image = image[y:y+h,x:x+w]

    #To handle those cases where the  height and width of the generated cropped face become 0
    if face_image.shape[0] and face_image.shape[1]:

      face_image = tf.image.resize(face_image, [image_w, image_h])
      face_image = face_image/127.5-1

      _cls = model.predict(np.expand_dims(face_image,axis=0))
      _cls = np.argmax(_cls,axis=1)

      text = '{}'.format(class_map[_cls[0]])

      patch = plt.Rectangle([x, y], w, h, fill=False,
                            edgecolor=color_map_image[_cls[0]], linewidth=linewidth)
      ax.add_patch(patch)
      ax.text(x, y, text, bbox={'facecolor':color_map_image[_cls[0]], 'alpha':0.2},
          clip_box=ax.clipbox, clip_on=True)
```

Proposed Model for Face Mask Detection:

On any given test image of a crowd-based setting of people, our final mask detection model runs as follows: Apply the RetinaNet Face model for face detection to generate detected face crops from the input image. Xception model for classification into mask and no-mask categories for the detected face is applied upon the detections generated by RetinaNet model. The final output of these two would be the faces detected by RetinaNet along with the predicted category for each face, that is whether the subject is wearing a mask or not.

Testing on images of densely crowded places

```
detector = RetinaFace(backbone='RESNET50')

image_path = '/content/CrowdMaskDetection/*'

for i in glob(image_path):
  image = cv2.cvtColor(cv2.imread(i), cv2.COLOR_BGR2RGB)

  result = detector.detect(image)

  boxes = []
  for i in range(len(result)):
    boxes.append(result[i]['box'])
  boxes = np.array(boxes)

  visualize_detections(image, boxes)
```

Testing on videos of densely crowded places

```
video_source = "/content/crowd-people-using-mask-video-id1213759078"

cap = cv2.VideoCapture(video_source)
img_array = []

while True:
  _, image = cap.read()
  if not _:
    break
  w, h = image.shape[0], image.shape[1]
  result = detector.detect(image)

  boxes = []
  for i in range(len(result)):
    boxes.append(result[i]['box'])
  boxes = np.array(boxes)
  img=visualize_detections_video(image, boxes)
  img_array.append(img)

out = cv2.VideoWriter('/content/mask.avi', 0, 24, (h,w))

for i in range(len(img_array)):
  out.write(img_array[i])

out.release()
```

Results and Discussion

- Results obtained on the classification model:

	precision	recall	f1-score	support
0	0.99	0.99	0.99	807
1	0.99	0.99	0.99	809
accuracy			0.99	1616
macro avg	0.99	0.99	0.99	1616
weighted avg	0.99	0.99	0.99	1616

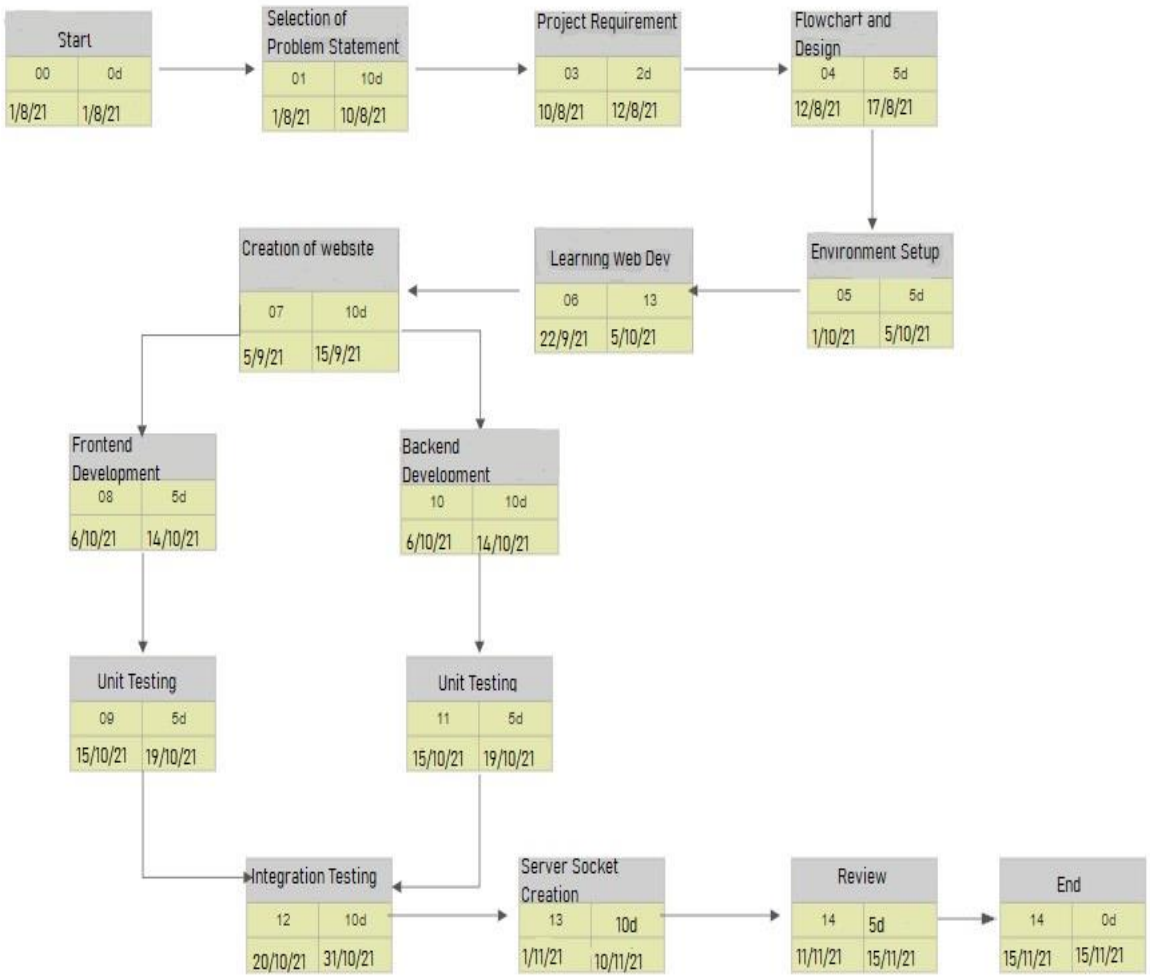
- Results on test images:


With mask
With mask
Without mask
LIVE


With mask
With mask
With mask
With mask
Without mask
With mask
Without mask
Without mask
Without mask
Without mask
Without mask
With mask


With mask
Without mask

PERT CHART



9) Pert Chart

CONCLUSION

Using the before mentioned approach of face detection and classification of the detected face crops as wearing a mask or not works pretty well in crowded conditions. This is really important because the use cases for the regions of surveillance may include metropolitan complexes, metro stations and dense marketplaces. These conditions do not provide an ideal scenario for just any face detection algorithm, and it was really necessary that the right choice was made.

Choice of RetinaNet over MT-CNN, Haar Cascade and HOG:

- While deciding upon the face detection algorithm to be used as part of my proposed solution of face detection, a pre-trained RetinaNet was chosen as the one which could, with the highest recall and precision, predict the number of faces in a crowded setting.

- In an uncontrolled environment, accurate face localization remains a challenge, and I needed a model which could efficiently predict, with a very high level of certainty- the people who are not wearing a mask in a crowded setting for the mask detection algorithm to work effectively.

- While they work good in general settings, MT-CNN (Multi Task Cascaded Convolutional Neural Network) and classical computer vision algorithms such as Haar Cascade failed to work well in an uncontrolled environment of a crowded setting with people and in dense clusters. RetinaNet with a ResNet backbone and feature pyramid network for feature extraction works even well than some single shot detectors like Single Shot MultiBox Detector and has accuracy on par with two stage detectors like Faster RCNN, and handles foreground-background class imbalance using a modified version of Focal Loss. The class imbalance was the major issue in other single shot detectors, and helps RetinaNet have a lower loss due to easy examples while focusing on hard ones.

- This face detection model works well in crowded settings, which are bound to have large number of people with 'smaller' faces and with varying scale than in a general setting. On the testing examples, it can be seen that the

faces detected vary in scale but are detected with a very high recall and precision.

Classification model: Choice of architecture

Transfer learning was employed to use the model weights of the Xception model. Learning hyperparameters such as learning rate were chosen in an iterative manner, with recommendations taken on choices of values based on available architecture.

Criterion for evaluation: F1-score

- F1-score is the harmonic mean of precision and recall. It is chosen as the criterion for evaluation for the classification model. Being bound between 0 and 1, F1-score reaches its best value at 1 and worst at 0.
- My model achieves a high F1-score which shows that it can perform well in real world scenarios to classify with certainty, the mask and without mask categories on face crops.

REFERENCES

- Chollet, Francois. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. 1800-1807. 10.1109/CVPR.2017.195.
- He, Kaiming & Zhang, Xiangyu & Ren, Shaoqing & Sun, Jian. (2016). Deep Residual Learning for Image Recognition. 770-778. 10.1109/CVPR.2016.90.
- Deng, Jiankang & Guo, Jia & Zhou, Yuxiang & Yu, Jinke & Kotsia, Irene & Zafeiriou, Stefanos. (2019). RetinaFace: Single-stage Dense Face Localisation in the Wild.
- Padilla, Rafael & Filho, Cicero & Costa, Marly. (2012). Evaluation of Haar Cascade Classifiers for Face Detection.
- S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.
- L. Breiman. Random forests. Machine learning, 45(1):5–32, 2001.
- D. Cire¸san, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. Arxiv preprint arXiv:1202.2745, 2012.
- D.C. Cire¸san, U. Meier, J. Masci, L.M. Gambardella, and J. Schmidhuber. High-performance neural networks for visual object classification. Arxiv preprint arXiv:1102.0183, 2011.
- J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In CVPR09, 2009.
- J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. Fei-Fei. ILSVRC-2012, 2012. URL http://www.image-net.org/challenges/LSVRC/2012/.
- L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. Computer Vision and Image Understanding, 106(1):59–70, 2007.
- G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007. URL http://authors.library.caltech.edu/7694.

- G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R.R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580, 2012.
- K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In International Conference on Computer Vision, pages 2146–2153. IEEE, 2009.
- A. Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto, 2009.
- A. Krizhevsky. Convolutional deep belief networks on cifar-10. Unpublished manuscript, 2010. [14] A. Krizhevsky and G.E. Hinton. Using very deep autoencoders for content-based image retrieval. In ESANN, 2011.
- Y. Le Cun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, et al. Handwritten digit recog
- J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.
- L. Calavia, C. Baladrón, J. M. Aguiar, B. Carro and A. Sánchez-Esguevillas, "A Semantic Autonomous Video Surveillance System for Dense Camera Networks in Smart Cities", *Sensors*, vol. 12, no. 8, pp. 10407-10429, Aug. 2012.
- Islam, M. Z., Islam, M. M., & Asraf, A. (2020). A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images. *Informatics in medicine unlocked*, *20*, 100412.
- Liu, L., Ouyang, W., Wang, X. *et al.* Deep Learning for Generic Object Detection: A Survey. *Int J Comput Vis* **128,** 261–318 (2020). https://doi.org/10.1007/s11263-019-01247-4
- https://github.com/X-zhangyang/Real-World-Masked-Face-Dataset
- https://www.kaggle.com/koyomi455/mask-dataset
- https://github.com/X-zhangyang/Real-World-Masked-Face-Dataset
- https://www.kaggle.com/ashishjangra27/face-mask-12k-images-dataset