PROJECT REPORT
CS 5542 BIG DATA ANALYTICS AND APPLICATIONS
OBJECT DETECTION AND ALERT SYSTEM

Course Coordinator - Yugyung Lee
Instructor - Syed Jawad Hussain Shah

Submitted by - Group 12
Vyoma Desai - 16314631
Tanvi Jain - 16270942
Mohammed Shaik - 16319967
Divyanshi Kothari - 12574897

ABSTRACT :

Object detection is one of the most interesting topics nowadays in computer vision. It mainly involves finding all the specific images and determining its object category and location. It is widely used and has greater value.

KEYWORDS :

Lightweight machine learning model, MobileNet SSD, Triangular Similarity, Wider Face Dataset, Alert System, gTTS.

INTRODUCTION:

There are 285 million people in the world who are visually imparied and it is very common that they suffer from regular and constant challenges in moving around, or going out alone especially in navigation when they are completely on their own and rely on someone else for assistance or help. These are the people who suffer the most in their day-to-day life and it becomes crucial for them to roam alone. So it is literally a challenging task and we have derived a strong technological solution for them which will help them to navigate with confidence and of course it is of utmost importance.

PURPOSE: In this project our sole idea is to develop a system that can guide visually challenged people to navigate on their own with confidence and also give them precise information by using alert systems which generate voice response/feedback and calculate distance which produces warnings. We have mainly focused/targeted on person as an object detection and along with this most challenging part is to calculate the accurate distance of people which is our key unique feature. Aim of this project is to develop an efficient lightweight model and a low cost system to help blind victims to navigate with greater confidence, comfort and with good speed.

DATASET :

One of the challenging subjects in computer vision is object detection, which helps organizations understand and recognise real-time objects with the help of digital pictures as inputs. We choose 32,203 images and label 393,703 faces with high degree of variability, in scale, pose and occlusion as depicted in the sample images.
We have worked on the following :

1. Research Face Dataset.

There are many available face detection dataset on the web, but after comparing it with many different dataset, we found WIDER FACE DATASET as much appropriate to our project.

2. Statistics.

   WIDER Face dataset is organized based on 61 event classes. For each event class, we have randomly selected 40%/10%/50% data as training, validation and testing sets.

3. Cleaning and Preprocessing manually to eliminate bad quality, poses, weird angles.

PROJECT OVERVIEW :

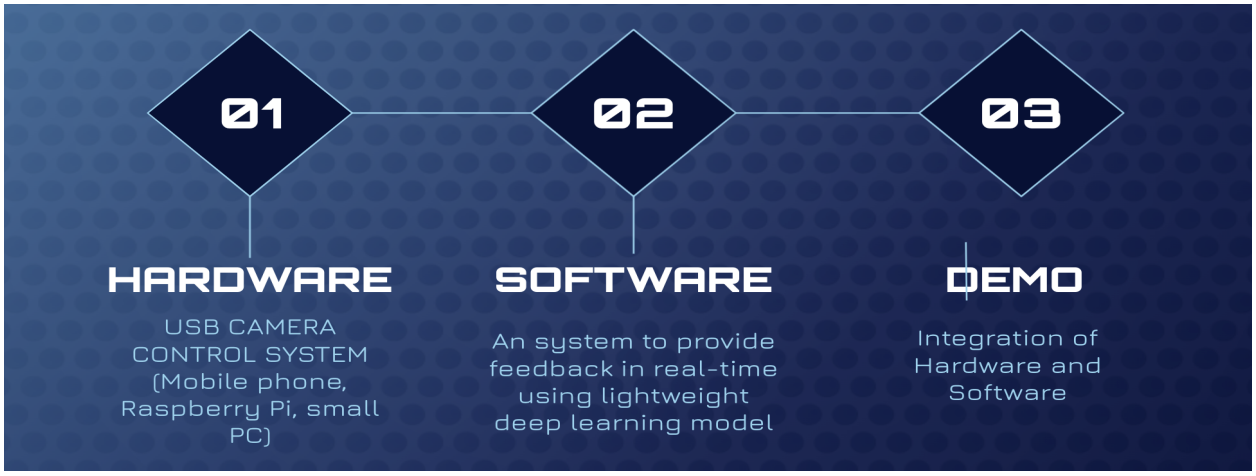

FIG 1 : PROJECT OVERVIEW FLOW DIAGRAM

HARDWARE CONSTRUCTION :

Main idea was to cover the entire 360 degree camera view, therefore we have at least set up 2 web cameras , one at front and other at back for this project. Front camera will cover the front

view and the back camera will cover the back view. For considering both views we used caps and this can be setup on hats too. Or we can just have a single 360 view camera and install at the center of the caps/hats.

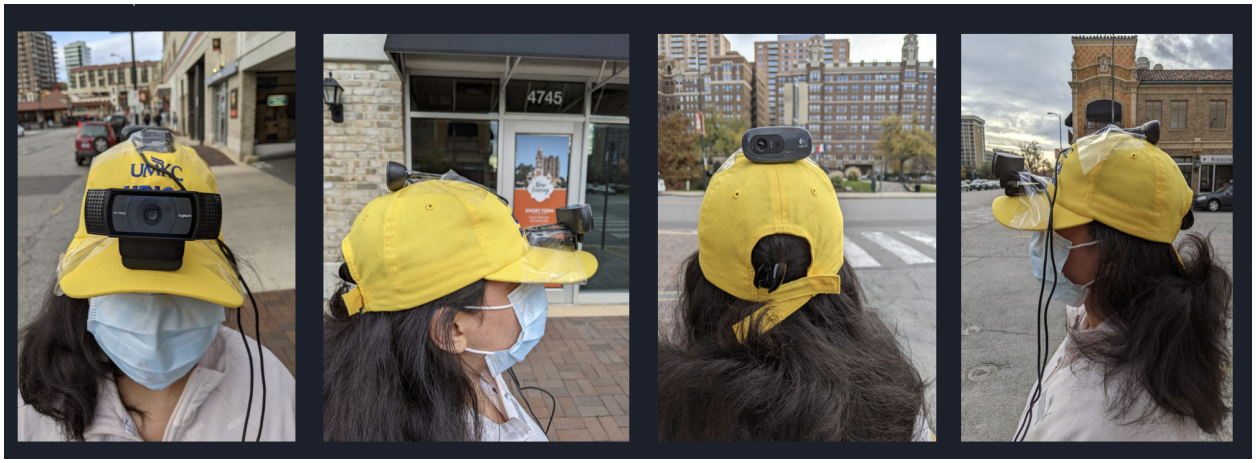

FIG 2 : HARDWARE CONSTRUCTION

SOFTWARE :

- **MACHINE LEARNING MODEL**

We developed a lightweight face detector using MobileNet SSD (Single Shot Detection) by using Tensorflow object detection API. This gives us inference time less than 7 milliseconds with an input size of 300 and 300 (image pixel size).

- **MobileNet SSD**

SSD - Single Shot Detection is one of the powerful models in detecting objects with greater accuracy. As the term SSD is called single shot detection where we have a single convolution layer that identifies the bounding box locations and at one go it can classify these locations. SSD network consists of base architecture i.e MobileNet with several convolutional layers.

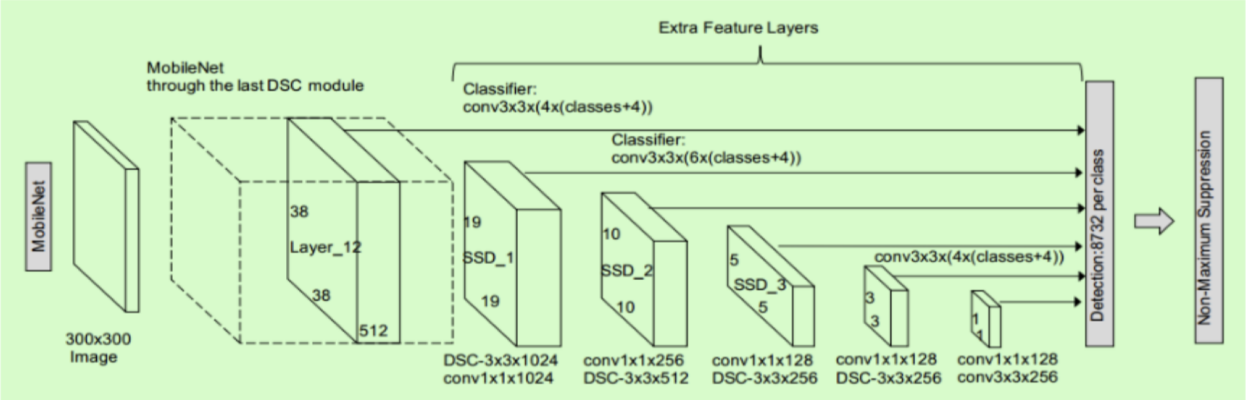

FIG 3 : MOBILENET SSD ARCHITECTURE

By using SSD, we can now detect and identify multiple objects in the image with the help of one single shot. There is a vast difference in SSD and RPN based models, RPN Regional Based Proposal approaches such as R-CNN require 2 shots, one for the region itself and one for detecting the object of every proposal. As we can see that SSD is much more time efficient as compared to any RPN based models. Therefore for our project we have built an SSD model for higher accuracy in detecting faces as an object.

- **TENSORFLOW OBJECT DETECTION API :**

To perform object detection problems we have used tensorflow api for creating a deep learning model.

- **Implementation :**

  Installations :

```
(tensorflow25) vyomas-mbp:~ vyo$ pip install tensorflow==2.2pip install --ignore-installed --upgrade tensorflow==2.2.0
(tensorflow25) vyomas-mbp:~ vyo$
(tensorflow25) vyomas-mbp:~ vyo$
(tensorflow25) vyomas-mbp:~ vyo$ pip install --ignore-installed --upgrade tensorflow==2.2.0
Collecting tensorflow==2.2.0
  Downloading tensorflow-2.2.0-cp38-cp38-macosx_10_11_x86_64.whl (175.4 MB)
     |████████████████████████████████| 175.4 MB 133 kB/s
```

```
(tensorflow25) vyomas-mbp:~ vyo$ python -c "import tensorflow as tf;print(tf.reduce_sum(tf.random.normal([1000, 1000])))"
2021-11-29 22:21:35.822182: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructions that this TensorFl
ow binary was not compiled to use: AVX2 FMA
2021-11-29 22:21:35.837748: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x7ff19d10c250 initialized for platform
Host (this does not guarantee that XLA will be used). Devices:
2021-11-29 22:21:35.837768: I tensorflow/compiler/xla/service/service.cc:176]   StreamExecutor device (0): Host, Default Version
tf.Tensor(468.45578, shape=(), dtype=float32)
```

```
(tensorflow25) vyomas-mbp:tensorflow vyo$ cd models/research/
(tensorflow25) vyomas-mbp:research vyo$ pwd
/Users/vyo/Desktop/tensorflow/models/research
(tensorflow25) vyomas-mbp:research vyo$ protoc object_detection/protos/*.proto --python_out=.
(tensorflow25) vyomas-mbp:research vyo$
```

```
(base) vyomas-mbp:research vyo$ git clone https://github.com/cocodataset/cocoapi.git
Cloning into 'cocoapi'...
remote: Enumerating objects: 975, done.
remote: Total 975 (delta 0), reused 0 (delta 0), pack-reused 975
Receiving objects: 100% (975/975), 11.72 MiB | 12.24 MiB/s, done.
Resolving deltas: 100% (576/576), done.
(base) vyomas-mbp:research vyo$ cd cocoapi/PythonAPI
(base) vyomas-mbp:PythonAPI vyo$ make
python setup.py build_ext --inplace
running build_ext
cythoning pycocotools/_mask.pyx to pycocotools/_mask.c
/Users/vyo/opt/anaconda3/lib/python3.8/site-packages/Cython/Compiler/Main.py:369: FutureWarning: Cython directive 'language_level'
 not set, using 2 for now (Py2). This will change in a later release! File: /Users/vyo/Desktop/tensorflow/models/research/cocoapi/
PythonAPI/pycocotools/_mask.pyx
  tree = Parsing.p_module(s, pxd, full_module_name)
building 'pycocotools._mask' extension
creating build
creating build/common
creating build/temp.macosx-10.9-x86_64-3.8
creating build/temp.macosx-10.9-x86_64-3.8/pycocotools
```

```
(tensorflow25) vyomas-mbp:research vyo$ cp object_detection/packages/tf2/setup.py .
(tensorflow25) vyomas-mbp:research vyo$ python -m pip install .
```

```
(tensorflow25) vyomas-mbp:research vyo$ cp object_detection/packages/tf2/setup.py .
(tensorflow25) vyomas-mbp:research vyo$ python -m pip install .
Processing /Users/vyo/Desktop/tensorflow/models/research
  DEPRECATION: A future pip version will change local packages to be built in-place without first copying to a temporary directory
. We recommend you use --use-feature=in-tree-build to test your packages with this new behavior before it becomes the default.
   pip 21.3 will remove support for this functionality. You can find discussion regarding this at https://github.com/pypa/pip/issu
es/7555.
Collecting avro-python3
  Downloading avro-python3-1.10.2.tar.gz (38 kB)
Collecting apache-beam
  Downloading apache_beam-2.34.0-cp38-cp38-macosx_10_9_x86_64.whl (4.3 MB)
     |████████████████████████████████| 4.3 MB 2.5 MB/s
Collecting pillow
  Downloading Pillow-8.4.0-cp38-cp38-macosx_10_10_x86_64.whl (3.0 MB)
     |████████████████████████████████| 3.0 MB 9.3 MB/s
Collecting lxml
  Downloading lxml-4.6.4-cp38-cp38-macosx_10_14_x86_64.whl (4.5 MB)
     |████████████████████████████████| 4.5 MB 40.5 MB/s
Collecting matplotlib
  Downloading matplotlib-3.5.0-cp38-cp38-macosx_10_9_x86_64.whl (7.3 MB)
     |████████████████████████████████| 7.3 MB 8.8 MB/s
Collecting Cython
  Using cached Cython-0.29.24-cp38-cp38-macosx_10_9_x86_64.whl (1.9 MB)
Collecting contextlib2
  Using cached contextlib2-21.6.0-py2.py3-none-any.whl (13 kB)
Collecting tf-slim
```

Using this API we trained mobilenet ssd face detector and achieved an accuracy of 86.39 % on our testing set.

The model was deployed in our demo to detect faces and the distance was estimated using triangle similarity as described below.

DISTANCE ESTIMATION CORE CONCEPT :

- **TRIANGLE SIMILARITY :**

We have created our own dataset which helps to calculate the distance of the object from the camera. This technique helps to find the distance from camera to object/marker using python and OpenCV. I have set up the equation in such a way that we have an object with known width W. We then place this marker at some distance D from our camera. Once I am at a particular distance then I will now take a picture of my own object using a camera and then calculate

apparent width P. By using the formula it will allow me to calculate the focal length F of our camera.

Focal length **F** = (Apparent width in Pixels **P** * Distance **D**) / Width of an object **W**

We have set up 2 web cameras so we need to calculate the focal length from each individual camera since each camera has a different pixel size value.

Unique Feature in our project:

Have we all noticed that the distance between our left eyebrow corner to right eyebrow corner ranges from 5.5 to 5.7 inch and this is usually the same for all humans. This value is fix and will be used as Width of the face object.

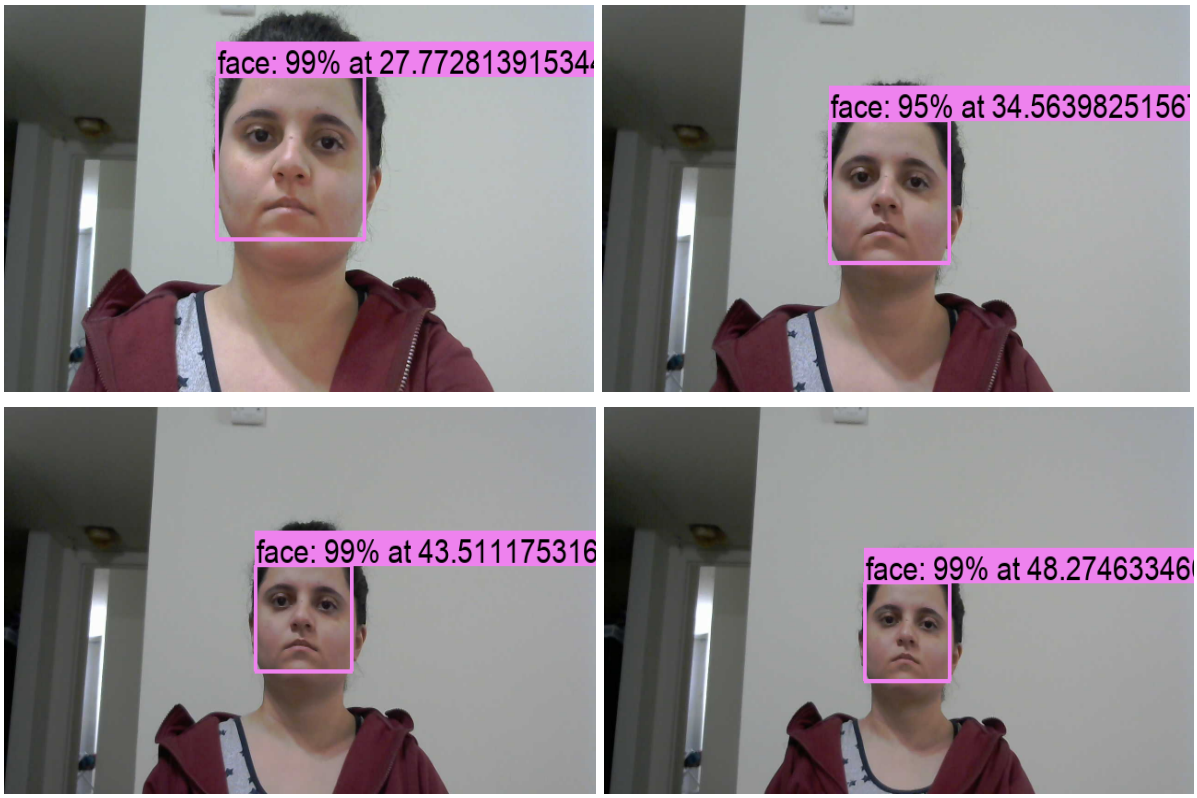

FIG 5 : Calculating Focal Length from 2 cameras front and back at different distances.

Web Camera 1 - Distance 1 (24 inch):

I was at a distance 24 feet away from the camera, my width in the image had perceived width as 156 pixels and my width is 5.7 inch. Therefore my focal length F is then :

F = (156px * 24in) / 5.7in = 656.84

So my focal length from camera 1 at a distance of 24 inch is 656.84.

Web Camera 2 - Distance 1 (24 inch):

Considering the same distance 24 feet, now I will calculate the focal length of camera 2 from the object. My width in the image had perceived width as 98 pixels and my left to right eyebrow width is 5.7 inch. Therefore my focal length F is then :

F = (98px * 24in) / 5.7in = 412.63
So my focal length from camera 2 at a distance of 24 inch is 412.63.

Now, I have calculated the focal length from both cameras at an equal distance of 24 inch, 30 inch, 36 inch, 42 inch, 48 inch, 54 inch and 60 inch. We now got an average focal length of camera 1 and camera 2 as 680 , 500 respectively. Once we have the average focal length of both the cameras, I can now apply triangle similarity concept to calculate distance of the object to the camera in real time.

Apparent Width in pixels P (image of object using camera)	Distance D (from the camera)	Width W (left to right human Eyebrow corner)	Focal Length F (of camera)
156 , 98	24	5.7	656.84 , 412.63
126 , 72	30	5.7	663.15 , 392.80
100 , 61	36	5.7	631.57 , 385.26
88 , 54	42	5.7	648.42 , 395.45
71 , 49	48	5.7	597.89 , 412.63
61 , 45	54	5.7	577.89 , 432.05

FIG 5 : TRIANGULAR SIMILARITY FOR CALCULATING FOCAL LENGTH - Camera 1 & 2

TEXT-TO-SPEECH CONVERSION SYSTEM :

This system is capable of sending alerts via text to speech mechanism by using the gTTS library. It will generate voice response/feedback which produces warnings whether the visually impaired person is very close or far away from the other approaching person at a particular distance in feets. Alerts are issued after every 5 seconds to avoid nuisance otherwise continuous voice feedback would be very irritating. Text-to-speech is triggered when the distance is less than 20 feet between both the person, this distance is good enough to alert the person for any approaching person. When in crowded scenes, we have focused

on saying 2 or more people to make him/her understand there are many people surrounding either from front or back or from both. This system is capable of sending the total number of people approaching from either front or back have also been implemented.

DEMO OF THE PROJECT (OUTCOME) :

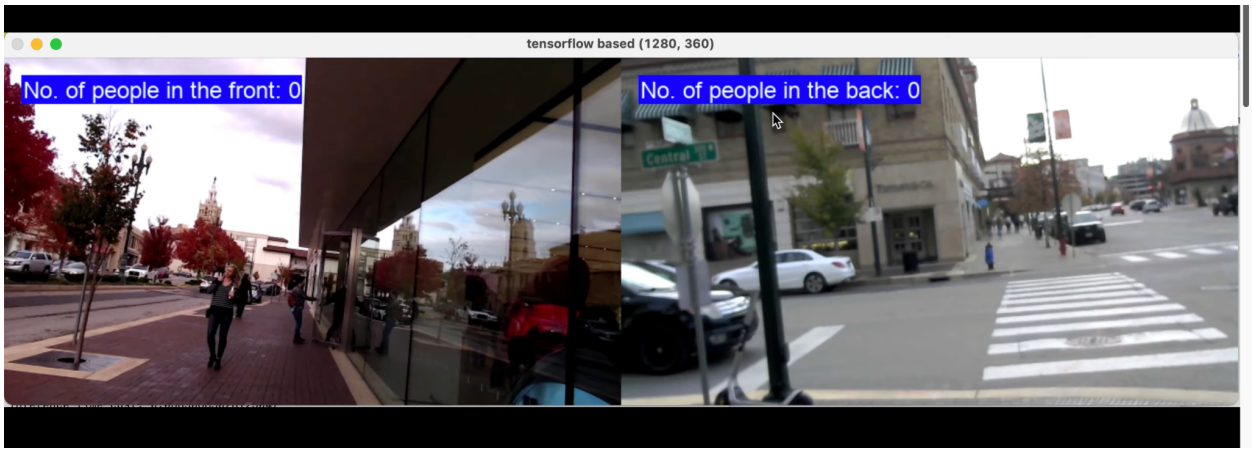

FIG 6 : DEMO This is the demo of our project which estimate distance in feets and count number of people

CURRENT FEATURES

Our system supports multiple cameras which provide a front and back view, which can help visually challenged people to go with confidence without any fear of collision with any object. Our MobileNet SSD model inference time is less than 7 milliseconds which is good enough to run on devices such as Raspberry Pi and mobile phones and other low cost hardware. Additional feature added Text-To-Speech Conversion. Using Triangle Similarity, Distance has been calculated from the camera to another person approaching either from front or at the back.

FUTURE WORK

1. To detect all kinds of object in addition to the pedestrians
2. Detecting the crosswalk light for safely crossing the road.
3. Detecting face masks on the pedestrians and issuing an appropriate alert.
4. Smart navigation can also be added.
5. Text-to-Speech can be replaced with a vibration system or other system.

REFERENCES :

1. Face Mask Detection: https://www.kaggle.com/andrewmvd/face-mask-detection

2. MaskedFace-Net: https://arxiv.org/abs/2008.08016
3. MobilenetSSD - https://medium.com/@techmayank2000/object-detection-using-ssd-mobilenetv2-using-tensorflow-api-can-detect-any-single-class-from-31a31bbd0691
4. MobilenetSSD https://medium.com/axinc-ai/mobilenetssd-a-machine-learning-model-for-fast-object-detection-37352ce6da7d
5. Object Detection TensorFlow https://towardsdatascience.com/object-detection-by-tensorflow-2-x-e1199558abc
6. GTTS https://towardsdatascience.com/how-to-get-started-with-google-text-to-speech-using-python-485e43d1d544

Our Github Link :
https://github.com/VyomaD/Hack-A-RooFall-2021

Our Youtube Video Link :
https://www.youtube.com/watch?v=v7FmaY1613U