IoT and Computer Vision for quantifying occupancy in a place

Team name:- Unbiased Enthusiasts

Team members:- Srinivasula Koushik (18277), Mallidi Poorna Siva Rama Reddy (18133), Desaraju Harsha Vardhan (18084)

***Abstract*—Human detection accuracy in a video surveillance system is critical for a variety of applications. For a growing number of complicated datasets, neural network techniques for image processing, classification, and detection are becoming more popular. Computers can be easily trained to recognize and categorize many items inside a picture with high accuracy using massive quantities of data, faster and more efficient GPUs, and improved algorithms. CNNs can do a better job at recognizing the number of people in a frame. In this project, we intend to detect the number of people in a given frame of a surveillance camera. We use CNN for this purpose. To train and test our full model, we considered a dataset composed of around 5700 images with the number of people in the images as labels in a separate excel file. We evaluate our method and demonstrate the results by comparing it with the MobileNetv3 model. MobileNetResv3 creates a feature map by running CNN on an image that is only supplied into the system from the live feed. Once done, the number of people detected in the given frame will be displayed as output.**

Keywords—Convolutional Neural Network, object detection, Deep learning, Computer Vision, (key words**)**

I. INTRODUCTION

Due to its wide variety of applications, recognizing human beings in a video segment of a surveillance system has gotten greater attention in recent years. Surveillance video sequences are often of poor quality. The majority of scenes caught by a stationary camera have little variation in the environment. The majority of current digital video surveillance systems rely on human observers to identify details about the scene, such as the number of persons present.

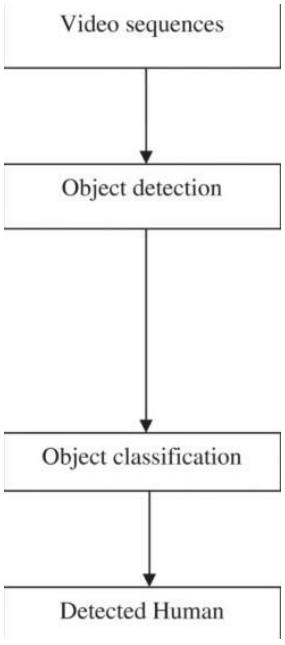
Motive force to watch multiple events via surveillance screens, however, has limitations. As a result, people identification in digital video surveillance has emerged as one of the most dynamic and interesting study subjects in computer vision and machine learning.

For accurate object detection, an expert system recognizes and collects target characteristics. The goal of this research is to detect humans. From a computer vision standpoint, human detection is a tough problem since it is impacted by a large range of potential appearances owing to changes in an articulated position, clothes, lighting, and backdrop, but an advanced understanding of these constraints can enhance recognition accuracy.

College sports arenas are the places of large gatherings and it is useful for the college community to know how busy the places are. It becomes necessary to know the occupancy of these places in order to take various measures like when to go there and it can be used for our new social norm that is social distancing.

To meet the above requirement, we need to have an end-to-end pipeline which starts with counting people in a given area and letting other users know about it.

In this project, we developed a system which is capable of giving the occupancy information in a place for the users. There are two stages for this project, first one is to develop an AI algorithm which can identify the occupancy count and second one is to pass on this information to a website and make it available to the public.



For first stage we have used 2 methods

- Developing an AI algorithm for counting the occupancy using CNN.

- Using a pretrained model for object detection.

Second model is used for comparative analysis

Second stage has 2 steps

- Capturing the live camera feed at the location of interest and sending it to the model frame by frame.
- Send predicted occupancy to the database
- Fetch this information from the database and then display it in the website developed.

II. CONTRIBUTIONS

Classification approach for object detection and integration of IoT and computer vision.
This project has a wide range of applications, ranging from

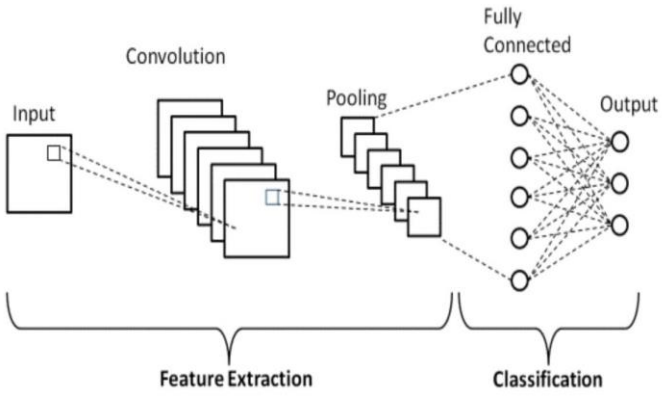
- Finding the occupancy of a place like sports arena, classroom, canteen, library etc.,
- Counting people in an open area,
- Implementation of COVID-19 protocols like social distancing, limiting the number of people in a given place.

The uniqueness of this project is that all the above-mentioned applications can be remotely monitored by the use of an online dashboard.

The goal of this project is to count people accurately in a location using an arbitrary still image and an arbitrary camera angle. At first glance, this appears to be an arduous undertaking since we need to overcome obstacles such as
1. Accurate object detection will not be an easy task with the unwanted objects in the image. Hence, we have to train a model and tune it to get better results and eliminate errors from unwanted objects in the foreground.
2. Because the scale of the people in the photos may vary significantly, we must combine characteristics from several scales to correctly predict people count for distinct images.

III. BACKGROUND

Convolutional Neural Network, commonly known as CNN is a powerful deep learning model that specializes in image-related tasks. The input to a CNN is a tensor with the following shape: (number of inputs) x (input height) x (input width) x (number of inputs) x (number of inputs) x (number of inputs) x (number of inputs) x (number of inputs) x (number of input (input channels). After passing through a convolutional layer, the image is abstracted into a feature map, also known as an activation map, with the following shape: (number of inputs) x (feature map height) x (feature map width) x (number of inputs) x (number of inputs) x (number of inputs) x (number of inputs) x (number of inputs) x (number of inputs) x (feature map channels).



The input is convolved by convolutional layers, which then pass the output to the next layer. This is analogous to a neuron's response to a single stimulus in the visual cortex. Each convolutional neuron only processes data for the receptive field it is assigned to. Although fully linked feedforward neural networks can be used to learn features and classify data, this architecture is unsuitable for bigger inputs like high-resolution photos. Along with standard convolutional layers, convolutional networks may add local and/or global pooling layers. By merging the outputs of neuron clusters at one layer into a single neuron at the next layer, pooling layers minimize the dimensionality of data. Small clusters are combined using local pooling, which typically uses tiling sizes of 2 x 2. The feature map's neurons are all affected by global pooling.

MobileNetV3 is a convolutional neural network that is tuned to mobile phone CPUs through a combination of hardware-aware network architecture search (NAS) complemented by the NetAdapt algorithm, and then subsequently improved through novel architecture advances. Depth wise separable convolutions are used by MobileNet. When correlated to a network with normal convolutions with the same depth in the nets, it dramatically reduces the number of parameters. As a consequence, lightweight deep neural networks are created. The MobileNet model, as its name implies, is intended for usage in smartphone platforms and is TensorFlow's first mobile computer vision model. MobileNet is a CNN class that was open-sourced by Google, and it provides us with an ideal starting position for building our ultra-small and ultra-fast classifiers. MobileNets are low-power, low-latency architectures that have been customized to match the resource restrictions of various use cases. Classification, detection, embeddings, and segmentation may all be constructed on top of them.

Input	Operator	exp size	#out	SE	NL	s
$224^2 \times 3$	conv2d, 3x3	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	✓	RE	2
$56^2 \times 16$	bneck, 3x3	72	24	-	RE	2
$28^2 \times 24$	bneck, 3x3	88	24	-	RE	1
$28^2 \times 24$	bneck, 5x5	96	40	✓	HS	2
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	120	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	144	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	288	96	✓	HS	2
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	conv2d, 1x1	-	576	✓	HS	1
$7^2 \times 576$	pool, 7x7	-	-	-	-	1
$1^2 \times 576$	conv2d 1x1, NBN	-	1024	-	HS	1
$1^2 \times 1024$	conv2d 1x1, NBN	-	k	-	-	1

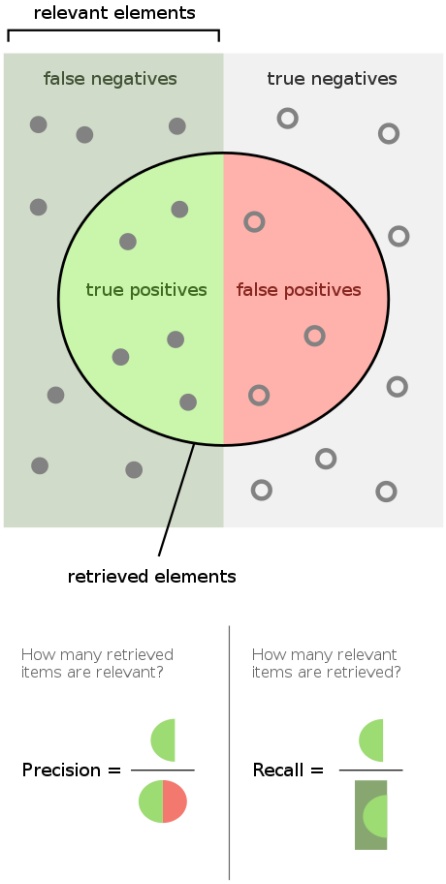
PostgreSQL is a powerful, open-source object-relational database system with over 30 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance. PostgreSQL is used as the primary data store or data warehouse for many webs, mobile, geospatial, and analytics applications.

Django is a high-level Python Web framework that encourages rapid development and clean pragmatic design. A Web framework is a set of components that provide a standard way to develop websites fast and easily. Django's primary goal is to ease the creation of complex database-driven websites. Some well-known sites that use Django include PBS, Instagram, Disqus, Washington Times, Bitbucket, and Mozilla.

Evaluation Criteria:

Accuracy: Model accuracy is the measurement used to determine which model is best at identifying relationships and patterns between variables in a dataset based on the input, or training, data.

Precision: Precision is one indicator of a machine learning model's performance – the quality of a positive prediction made by the model. Precision refers to the number of true positives divided by the total number of positive predictions (i.e., the number of true positives plus the number of false positives).



Recall: Recall literally is how many of the true positives were recalled (found), i.e. how many of the correct hits were also found. Precision (your formula is incorrect) is how many of the returned hits were true positive i.e. how many of the found were correct hits.

F1score: The F1-score is a metric for how accurate a model is on a given dataset. It's used to assess binary classification systems that divide examples into 'positive' and 'negative' categories. The F-score, which is described as the harmonic mean of the model's accuracy and recall, is a technique of integrating the model's precision and recall. The F-score is a popular metric for assessing information retrieval systems like search engines, as well as a variety of machine learning models, particularly in natural language processing. It's feasible to tweak the F1-score such that accuracy takes precedence over recall, or vice versa.

$$F_1 = \frac{2}{\frac{1}{\text{recall}} \times \frac{1}{\text{precision}}} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$
$$= \frac{\text{tp}}{\text{tp} + \frac{1}{2}(\text{fp} + \text{fn})}$$

IV. MATERIALS AND METHODS

Stage-1:

The problem of identifying the occupancy of a room or place can be divided into three parts:

- Sending the camera feed from the place of interest
- Processing the camera feed and identifying the number of people
- Updating this information in an online dashboard

Since step 1 concerns the onsite implementation of the project, it is not dealt with here. To implement step 2, we need a program that can predict the number of people in an image sampled from the camera feed. To do this, we need an AI model that can count the number of people in an image. This can again be further divided into 4 steps:

- Data Collection
- Data Preprocessing
- Model training
- Model Testing

The dataset we used consisted of around 5700 images of people and their counts. The dataset is small because of the limitations of computational resources. The preprocessing involved scaling all the photos to the same size of 256x256. We trained a Convolutional Neural Network (CNN) with this data. The details about the architecture are discussed in detail in the next section. After training, the model was tested against 30 images to evaluate the performance of the model. To know how well the model is performing, we compared it with the performance of a pre-trained model. This is to know where our model stands with respect to the benchmark model for this task which is MobileNetV3. A comparative analysis of this is discussed in the results section.

After the model is trained, it is saved for future use. A separate program for using this model and predicting the number of people in an image is written to use it in the webserver.

Stage-2: Combining AI and IoT.

In stage-1, our final output is occupancy count in a given place.

In this stage, we developed a pathway where this model from stage1 is deployed and output is made available for the public.

There are certain steps involved here:

Step-1:

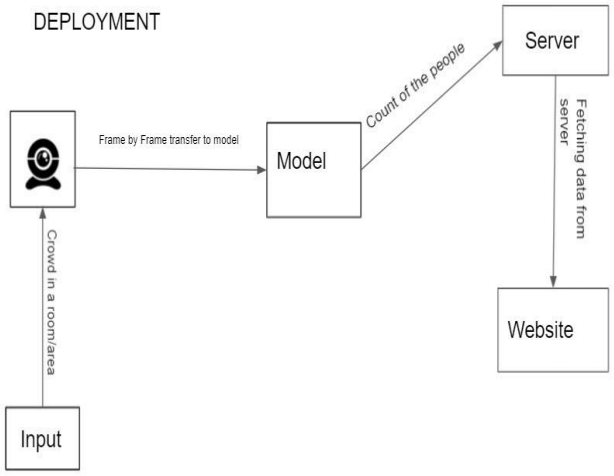
Through an external camera, a live camera feed of the place is captured and sent to the model frame by frame with a fixed frequency as the model can take only a single frame at a time.

Step-2:

For every frame sent, the model now can give the output of the number of people in the frame. The count is now sent to the database connected along with a timestamp.

Step-3:

A website is set up with this database as the backend. From the database, the latest information about the occupancy is fetched and displayed on the dashboard which is available for the public.



Prototype of the website:

Dataset:

This dataset consists of around 5700 training images with respective labels. Labels indicate the number of people in the frame. Labels are separately stored in the 'train.csv' file with the image name as column 1 and occupancy as column 2. Testing data consists of 30 images. 'Test.csv' consists of the file name as column 1 and occupancy as column 2.

Model Architecture:

We used a CNN model because CNNs specialize in image-related tasks. The first layer has the same shape as that of the image because it is the input layer. Next is a Con2D layer of size 64 and a kernel size of 3x3 with a RelU activation function, followed by a MaxPool layer of size 2x2. After that is a layer similar to the previous layer but of double the size i.e., of size 128. Then a dropout layer is added with a dropout rate of 0.2. Then the nodes are flattened to input them to a dense layer of size 128. Then another dense layer with 7 nodes is added followed by a dense layer of size 1, which is the output layer. The network has a total of around 63 million parameters, all of which are trainable.

Layer (type)	Output Shape	Param #
conv2d_2 (Conv2D)	(None, 254, 254, 64)	1792
max_pooling2d_2 (MaxPooling 2D)	(None, 127, 127, 64)	0
conv2d_3 (Conv2D)	(None, 125, 125, 128)	73856
max_pooling2d_3 (MaxPooling 2D)	(None, 62, 62, 128)	0
dropout_1 (Dropout)	(None, 62, 62, 128)	0
flatten_1 (Flatten)	(None, 492032)	0
dense_3 (Dense)	(None, 128)	62980224
dense_4 (Dense)	(None, 7)	903
dense_5 (Dense)	(None, 1)	8

Total params: 63,056,783
Trainable params: 63,056,783
Non-trainable params: 0

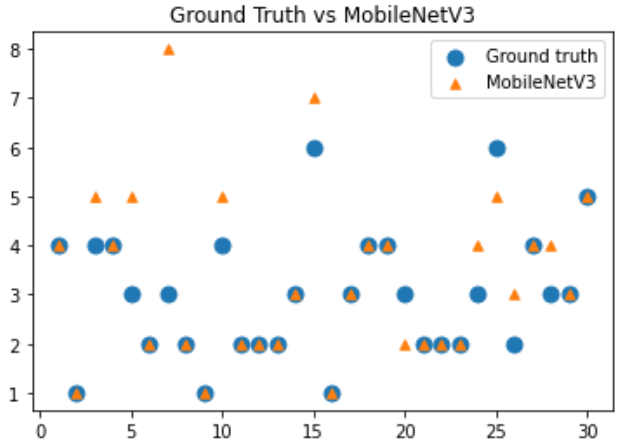
The model was trained for 50 epochs with a batch size of 16. Callback was also used for the learning rate to increase the speed of convergence. Despite using a small training size, the model had an accuracy of 46.67% whereas MobileNetV3 which is considered stats-of-art had an accuracy of 66.67% on the same test set. From this, we can conclude that the model has a decent performance which can be improved by training on a bigger dataset.

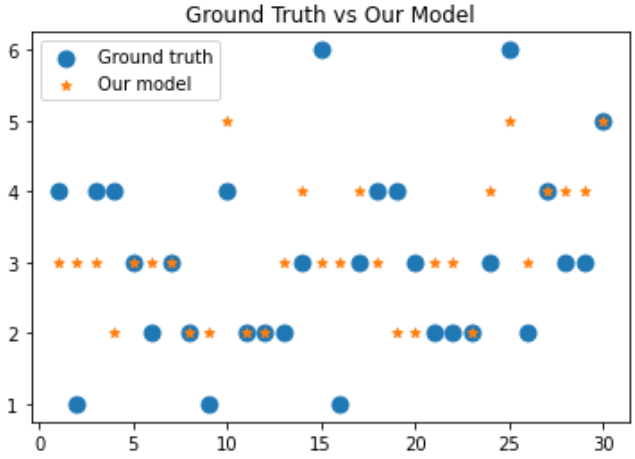
V. Results

Numerical analysis:

Accuracy of our model: 46.67%

Accuracy of pretrained model: 66.67%





A sample of comparative analysis is described in the above figure stating the predictions from our model and MobileNetV3. A detailed explanation of our views behind the results is given in the next section.

Our model	2	2	2	3
Mobilenet V3	2	4	2	2

Our model	2	2	2	3
Mobilenet V3	2	4	1	1

The below are the precision and recall scores our model and MobileNetV3

	Our Model	MobileNetV3
Precision	0.1923	0.4441
Recall	0.3062	0.4972
F1 Score	0.4295	0.2191

VI. Discussions

Some case studies:

Case Study-1:

In this case, both the models predicted perfectly as 2.

Our intuitive analysis for this case, in this photo, both the player's faces are clearly visible and therefore our model has been effective like the MobileNetV3.

Case Study –2:

In this picture, our model prediction was 4 as we can see the people in the back of the image are not so clear and it's tough to identify them with a model which is trained with images that are a little clear and all the people are relatively closer to the frame in the training data. But the pre-trained model has a well-developed complex architecture that can detect persons even in the background as it is trained on a bigger dataset and moreover MobileNetV3 is not classifying the images, es, it is trained in such a way that it identifies the human shape in a given picture irrespective of the number whereas our model classifies the image into labels which are nothing but the count of the people. But what can be observed is that this technique of classification can also work fine in overall prediction, it just needs a bigger dataset with more samples for each label and a variety of samples in each case.

There is a specific reason why we choose to go with classification rather than object detection.

That is because, this problem of occupancy is already answered with complex object detection algorithms and moreover the build of those algorithms from the scratch is computationally expensive and near to impossible with the resources we have, so if not from scratch, another idea is to implement transfer learning but again, usage of transfer learning is already an established answer for this problem and in transfer learning there is no innovation we can present.

So, we tried to tackle this problem with classification technique which turned out to be a fine approach but for sure as of now, object detection algorithms have an upper hand in dealing with occupancy problems.

Future Scope:

- As discussed in the above case study 2, we believe that having a more curated dataset is required if the plan is to identify the occupancy by the classification rather than object detection as it provides various kinds of samples which can enable the model to predict correctly even in the case of complex images like the one discussed in case study-2.
- Dashboard can be styled more with CSS for making it look more appealing and professional.

VII. Conclusions

Detecting human beings accurately in a surveillance video is one of the major topics of vision research due to its wide range of applications. It is challenging to process the image obtained from a surveillance video as it has low resolution.

We would like to conclude that, with a different approach we have managed to attain a decent accuracy in predicting the number of people in a given frame.

We have successfully managed to make a pathway where the predictions are stored in a database and later fetched to a website which can act as a dashboard.

The primary two objectives that are calculating occupancy and displaying the same in a website are met.

REFERENCES

[1] https://pjreddie.com/media/files/papers/YOLOv3.pdf

[2] https://personal.ie.cuhk.edu.hk/~ccloy/files/crowd_2013.pdf

[3] https://personal.ie.cuhk.edu.hk/~ccloy/files/bmvc_2012b.pdf

[4] https://personal.ie.cuhk.edu.hk/~ccloy/files/iccv_2013_crowd.pdf

[5] https://keras.io/api/applications/resnet/

[6] https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Zhang_Single-Image_Crowd_Counting_CVPR_2016_paper.pdf

[7] https://developpaper.com/brief-introduction-of-mobilenetv1-v2-v3-lightweight-network/

[8] https://www.postgresql.org/docs/current/

[9] https://docs.djangoproject.com/en/3.2/

[10] Human Detection in Surveillance Videos and Its Applications - A Review By: Manoranjan Paul, Shah M E Haque and Subrata Chakraborty, Paul et al. EURASIP Journal on Advances in Signal Processing 2013, Springer

[11] Multi-Class Moving Target Detection with Gaussian Mixture Part Based Model by: Jie Yang, Ya-Dong Sun, Mei-Jun Wu, and Qing-Nian Zhang, IEEE International Conference on Consumer Electronics (ICCE), 2014

[12] People Detection in Low-Resolution Video with Non-Stationary Background By: Jianguo Zhang, Shaogang Gong, Image and Vision Computing 27 (2009) 437–443, SciVerse ScienceDirect

[13] Human detection based on motion object extraction and head–shoulder feature by: Qing Ye, Rentao Gu, Yuefeng Ji, Optik 124 (2013) 3880– 3885, SciVerse ScienceDirect

[14] Fast Human Detection using Motion Detection and Histogram of Oriented Gradients By: Hou Beiping, Zhu Wen, JOURNAL OF COMPUTERS, VOL. 6, NO. 8, AUGUST 2011