Crowd Counting

- Anwesha Tomar

Introduction:

When there is an influx of a large number of people it can cause accidents such as a stampede. During sports events, large stadiums are full of people, this project can be used in order to analyse and predict the number of people in each area and prevent any accidents from taking place. It can also be used to analyse the number of people in a mall in order to prevent overcrowding and avoid mishaps from taking place .The relevance of this project can be seen in 2020, when there is an outbreak of a virus, it is mandatory to remain in doors, this project can be used to alert authorities when the number of people goes above a certain threshold. The goal of this project is to be able to be able to predict the number of people in each image.

In this report I describe my tasks and the results. My task was to ensure that data was loaded and pre-processed correctly. The other members built the model, while all three of us worked on finding the appropriate hyper parameters. Lastly, all three of us researched the Resnet50 model, I worked on data input for Resnet50, Wenyu built the model and Jingya found the appropriate hyper parameters.

Work done by me-

Dataset:

The dataset consists of a collection of 2000 images, each image has a size of 480x640 pixels and has 3 channels. The images were collected from a still webcam that is located inside a mall. Each image has a different number of people. We are provided with a csv file containing labels that give the number of people present in each image.

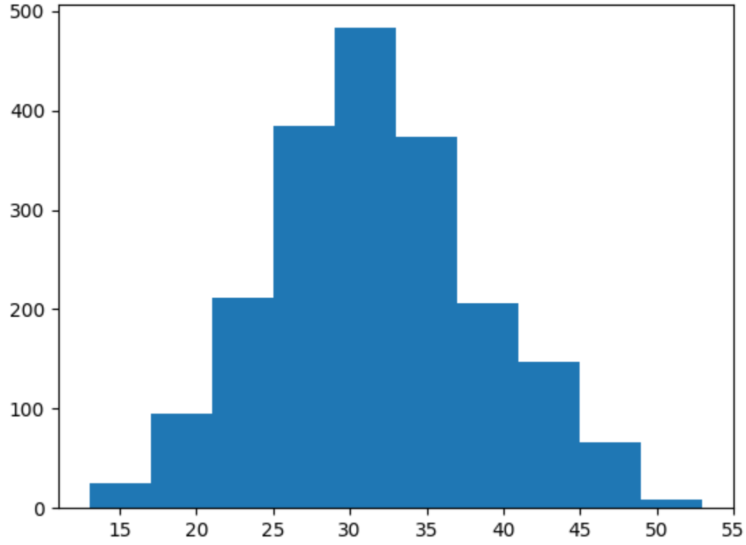


Fig 1. Distribution of labels

The image data is displayed below.

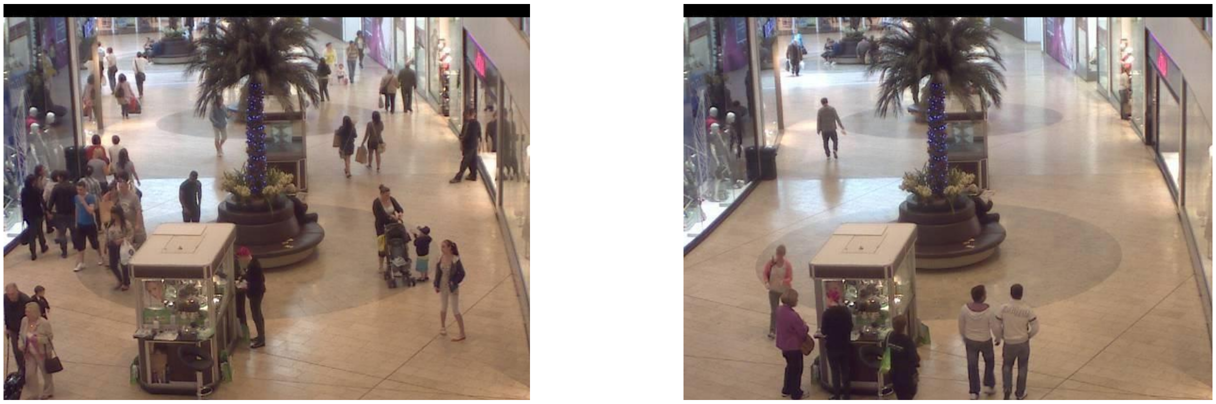
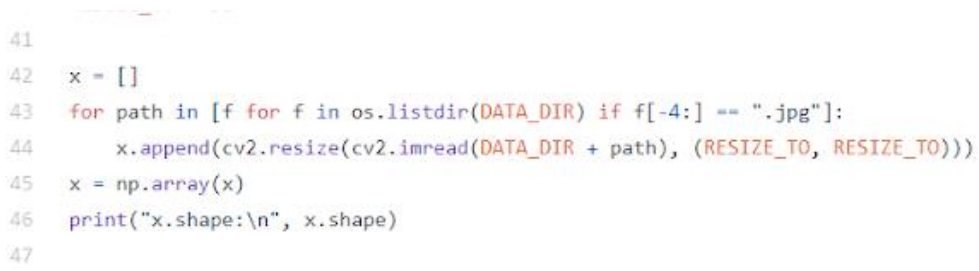


Fig 2. Image data that is input to the training models

Data Cleaning and Preprocessing:

The data is preprocessed and resized to 128x96 in order to store and train the data easily.



```python

x = []
for path in [f for f in os.listdir(DATA_DIR) if f[-4:] == ".jpg"]:
    x.append(cv2.resize(cv2.imread(DATA_DIR + path), (RESIZE_TO, RESIZE_TO)))
x = np.array(x)
print("x.shape:\n", x.shape)

```

Fig 3. Code to pre process data

Resnet 50:
Data inputs:- code snippet

```python
# %% ----------------------------------
--- Load Data --------------------------
----------------------------------------

DATA_DIR = os.getcwd() + "/frames/"

RESIZE_TO = 50

x, y = [], []

label = pd.read_csv("labels.csv")

label['image_name'] =
label['id'].map('seq_{:06d}.jpg'.format)
lb = label['id']

count = label['count']



for path in [f for f in
os.listdir(DATA_DIR) if f[-4:] ==
".jpg"]:

x.append(cv2.resize(cv2.imread(DATA_DIR
+ path), (RESIZE_TO, RESIZE_TO)))
```

```
    for j in range(len(lb)):
        if lb[j] == int(path[4:-4]):
            y.append(count[j])
x, y = np.array(x), np.array(y)
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=42, test_size=0.2)
x_train, x_test = x_train / 255, x_test / 255


resize = 64
batch = 256
# %% ------------------------------------------------- Data Prep------------------------------------------------------------------


# ImageDataGenerator - with defined augmentaions
datagen = ImageDataGenerator(
    rescale=1. / 255,
    featurewise_center=False,
    samplewise_center=False,
    featurewise_std_normalization=False,
    samplewise_std_normalization=False,
    zca_whitening=False,
    horizontal_flip=True,
    vertical_flip=True,
    validation_split=0.2,



preprocessing_function=resnet50.preprocess_input
)
```

```
flow_params = dict(
   dataframe=label,
   directory='/home/anwesha/ml2-project/frames',
   x_col="image_name",
   y_col="count",
   weight_col=None,
   target_size=(resize, resize),
   color_mode='rgb',
   class_mode="raw",
   batch_size=batch,
   shuffle=True,
   seed=0
)


# The dataset is split to training and validation sets
train_generator = datagen.flow_from_dataframe(
   subset='training',
   **flow_params
)
valid_generator = datagen.flow_from_dataframe(
   subset='validation',
   **flow_params
)


batch = next(train_generator)
```

Conclusion:

We can see that the ResNet50 model performed better than the CNN model, but there is still room for improvement. In future we would look to collect more data and improve the performance of our model. As of now we are limited because the data collected does not have any images for more than 55 people, our next step would be to find such data points.

Percentage of code from the internet: ((157-28)/(157+40))*100=64%

References:

1. https://www.researchgate.net/figure/Left-ResNet50-architecture-Blocks-with-dotted-line-represents-modules-that-might-be_fig3_331364877
2. https://www.kaggle.com/shovalt/crowd-counting-keras-pretrained-resnet50-cnn
3. https://www.kaggle.com/rmishra258/counting-crowd-with-cnn-social-distancing-project
4. https://www.mathworks.com/help/deeplearning/ref/resnet50.html#:~:text=ResNet%2D50%20is%20a%20convolutional,%2C%20pencil%2C%20and%20many%20animals.