**THEFT PREVENTION USING AI**

**a)Neural Network:**

Keras and Tensorflow are used in this project , Keras is a deep learning framework to enable fast experimentation with deep learning Keras is a neural network library while TensorFlow is the open-source library for a number of various tasks in machine learning. TensorFlow provides both high-level and low-level APIs while Keras provides only high-level APIs.Both frameworks thus provide high-level APIs for building and training models with ease here sklearn neural network ,it is used for standard machine learning operations such as model evaluation and model hyperparameter optimization can save a lot of time over implementing these schemes yourself.

**b)Optimizer:**

SGD(Stochastic gradient descent) is much faster but the convergence path of SGD is noisier than that of original gradient descent. SGD takes a lot of update steps but it will take a lesser number of epochs the number of times we iterate through all examples will be lesser in this case and thus it is a much faster process. Adam optimizer is chosen the method is really efficient when working with large problem involving a lot of data or parameters. It requires less memory and is efficient .

**c)Accuracy Metrics:**

Accuracy class is used here calculates how often predictions equal labels.This metric creates two local variables, total and count that are used to compute the frequency with which y\_pred matches y\_true. This frequency is ultimately returned as binary accuracy: an idempotent operation that simply divides total by count.

**d)Loss:**

Categorical crossentropy is a loss function that is used in multi-class classification tasks. These are tasks where an example can only belong to one out of many possible categories, and the model must decide which one. Formally, it is designed to quantify the difference between two probability distributions.

**e)Data Preprocessing:**

Data Preprocessing is done byfirst get the dataset,

• and then import libraries,importing the dataset,

• read csv function,

• extracting dependent and independent variable,

• handling missing data,

• encoding the categorical data,

• splitting the data set into train and test set,

• and feature scaling

**f)Right Shape:**

Getting your data into the right shape is important in ensuring accurate results when using Machine Learning. Let's investigate how to help make sure you have that shape.As far as your image size is concerned. It depends upon your problem and the original quality of the image. As you decrease the size of the image it leads to information loss. So, as long as your model performance does not degrades you can keep on reducing the size of the image but their will come a point where decrease in images size will lead decrease in model accuracy. That size is the threshold and you don't want to go below it.reshaping is done

**g)Function and features of open cv:**

\*changing the color spaces-gray\_image=cv2.cvtColor(image,cv2.COLOR\_BGR2GRAY)

\*edge detection- edges = cv2.Canny(image,100,200)

\*facedetection- for (ex,ey,ew,eh) in eyes: cv.rectangle(roi\_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)

\*plot the image- plt.imshow(img)

**h)Dataset:**

For known person we have used the dataset of any one of the known face, in my case i used known face for **“BARAK OBAMA”** and for unknown faces we have trained our model with the dataset of various different type of faces.

Here is the link for code file in .ipynb extension

<https://drive.google.com/file/d/1B7b_vXWv5hU8unRYCgE64QeozK19yJJn/view?usp=sharing>