- **Project description: (AI retail cashier)**

Instead of scanning products by QR code, by using computer vision we can build machine models that can identify the product itself which facilitates the shopping experience

- **Project planning**

1. **We chose the CNN model algorithm for this project as it is particularly well-suited for image recognition.**
2. **We found this data on Kaggle that is a collection of images of retail products like water bottles, noodles, chips and others.**

- **Dataset**

  Data is divided into 6 categories:

  1. **Aqua**

  2. **Chitato**

3. **Indomie**

4. **Pepsodent**

5. **Shampoo**

6. **Tissue**

code overview

We had to make 5 models to reach the best model fit and accuracy.

- **First data preprocessing**

```
# Scaling our data
trdata= ImageDataGenerator(rescale=1./255)
traindata=trdata.flow_from_directory(directory=train_loc, target_size=(224,224))
```

```
Found 237 images belonging to 6 classes.
```

```
tsdata= ImageDataGenerator(rescale=1./255)
testdata=tsdata.flow_from_directory(directory=test_loc, target_size=(224,224))
```

```
Found 60 images belonging to 6 classes.
```

```
[ ] traindata.class_indices
```

```
{'aqua': 0,
 'chitato': 1,
 'indomie': 2,
 'pepsodent': 3,
 'shampoo': 4,
 'tissue': 5}
```

We scaled the data by dividing by 255 and converted the class labels into numerical values from 0 to 5

- **Second model_1:**
  from os import name
  input_shape=(224,224,3)
  #input_layer
  Img_input=Input(shape=input_shape,name="this-the_input_layer")

  #conv layers

  x=Conv2D(32,(3,3),padding='same',activation='relu',name='layer_1')(Img_input)

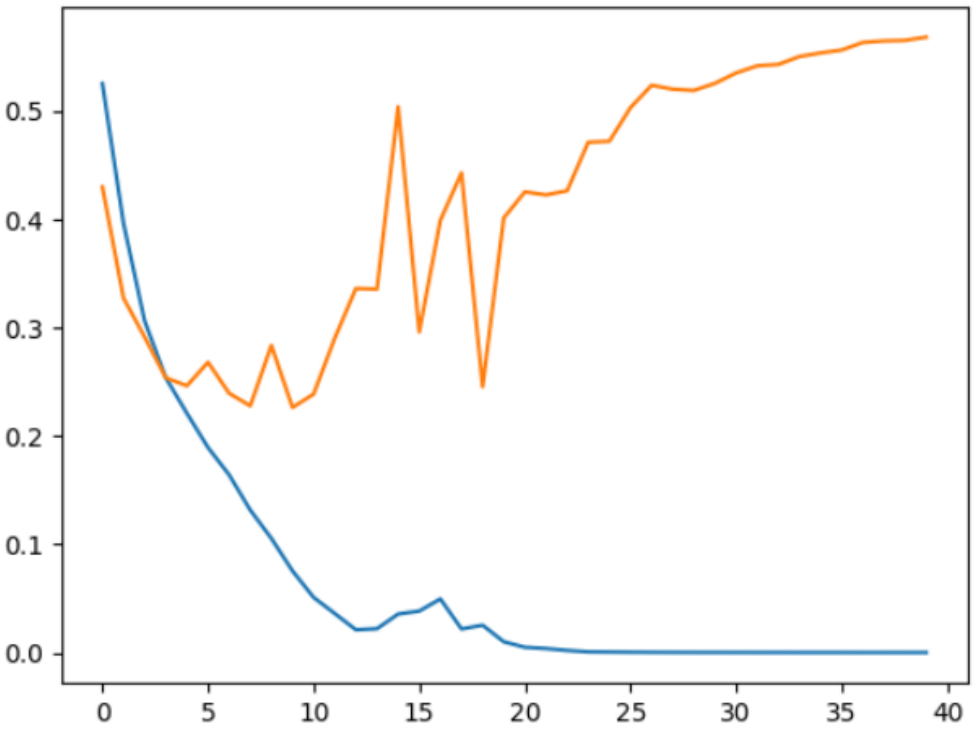
```
x=Conv2D(64,(3,3),padding='same',activation='relu',name='layer_
2')(x)
x=MaxPool2D((2,2),strides=(2,2),name='maxplool1')(x)
# x=Dropout(0.25)(x)

x=Conv2D(32,(3,3),padding='same',activation='relu',name='layer_
3')(x)
x=MaxPool2D((2,2),strides=(2,2),name='maxplool2')(x)

x=Conv2D(32,(3,3),padding='same',activation='relu',name='layer_
4')(x)
x=MaxPool2D((2,2),strides=(2,2),name='maxplool3')(x)

x=Flatten(name='flat')(x)
x=Dense(64,name='classifer')(x)
# x=Dropout(0.25)(x)
x=Dense(6,activation='softmax',name='classification')(x)
```

1) model summary:

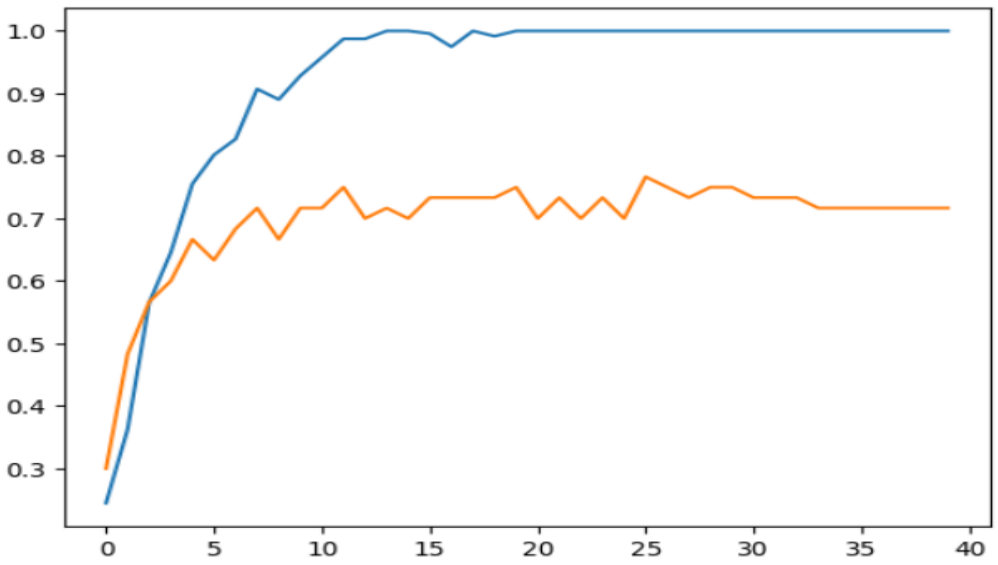
```
model_1.summary()
```

```
Model: "Cashier_classification"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 this-the_input_layer (Input  [(None, 224, 224, 3)]    0
 Layer)

 layer_1 (Conv2D)            (None, 224, 224, 32)      896

 layer_2 (Conv2D)            (None, 224, 224, 64)      18496

 maxplool1 (MaxPooling2D)    (None, 112, 112, 64)      0

 layer_3 (Conv2D)            (None, 112, 112, 32)      18464

 maxplool2 (MaxPooling2D)    (None, 56, 56, 32)        0

 layer_4 (Conv2D)            (None, 56, 56, 32)        9248

 maxplool3 (MaxPooling2D)    (None, 28, 28, 32)        0

 flat (Flatten)              (None, 25088)             0

 classifer (Dense)           (None, 64)                1605696

 classification (Dense)      (None, 6)                 390

=================================================================
Total params: 1,653,190
Trainable params: 1,653,190
Non-trainable params: 0
```

2) model losses (blue line for training and yellow for test , no. of epochs = 40)



3) Model accuracy



4) Observation

Over fitting model as the train accuracy is 1 and test acc is 0.7 and the test accuracy is low.

- **Third model_2**

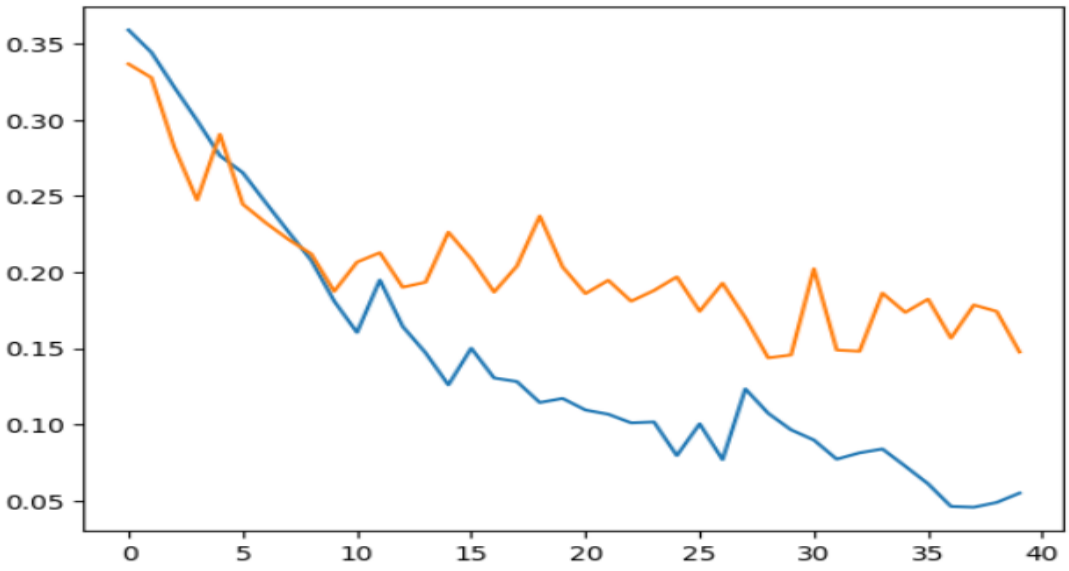
  We added the training data by applying augmentation and added dropout to the model to prevent overfitting problem.

  trdata_2= ImageDataGenerator(rescale=1./255,shear_range=0.2, zoom_range=0.2,horizontal_flip=True,vertical_flip=True, rotation_range=10)
  traindata_2 = trdata_2.flow_from_directory(directory=train_loc, target_size=(224,224), class_mode='categorical')

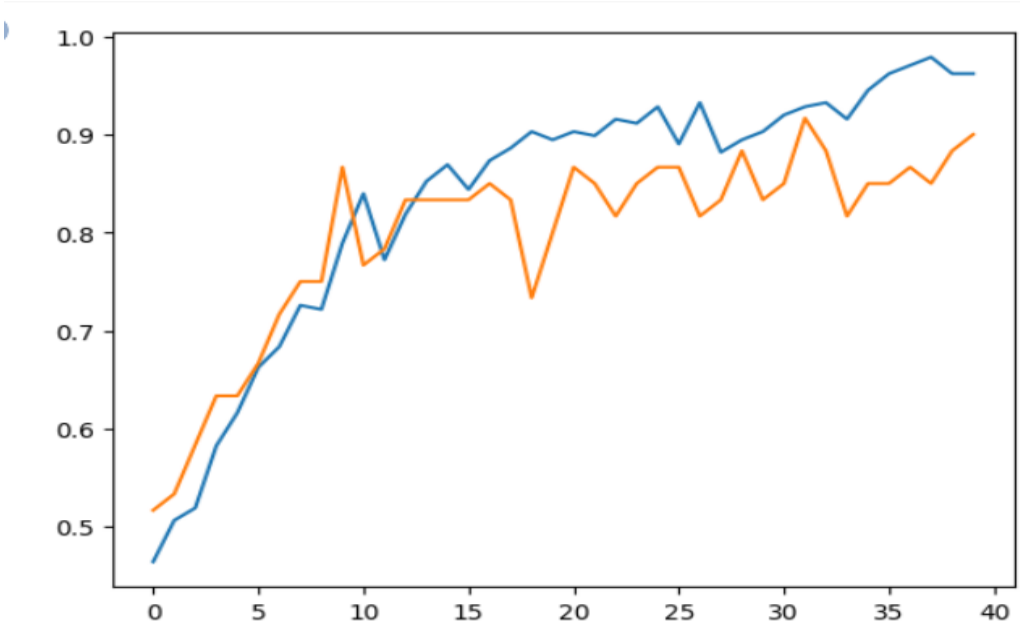
1. **Model summary**

```
Model: "Cashier_classification"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 this-the_input_layer (Input  [(None, 224, 224, 3)]    0
 Layer)

 layer_1 (Conv2D)            (None, 224, 224, 32)      896

 layer_2 (Conv2D)            (None, 224, 224, 64)      18496

 maxplool1 (MaxPooling2D)    (None, 112, 112, 64)      0

 layer_3 (Conv2D)            (None, 112, 112, 32)      18464

 maxplool2 (MaxPooling2D)    (None, 56, 56, 32)        0

 layer_4 (Conv2D)            (None, 56, 56, 32)        9248

 maxplool3 (MaxPooling2D)    (None, 28, 28, 32)        0

 flat (Flatten)              (None, 25088)             0

 classifer (Dense)           (None, 64)                1605696

 dropout (Dropout)           (None, 64)                0

 classification (Dense)      (None, 6)                 390

=================================================================
Total params: 1,653,190
Trainable params: 1,653,190
Non-trainable params: 0
_________________________________________________________________
```

2. Model losses



3. Model accuracy



4. **Observation**

   Train acc is 96 %and test acc is 90%

   the test accuracy has increased and the over fitting decreased but still can be improved

- **Fourth model_3**

  In the last model we had high variance problem (overfitting) so in order to fix it we applied regularization

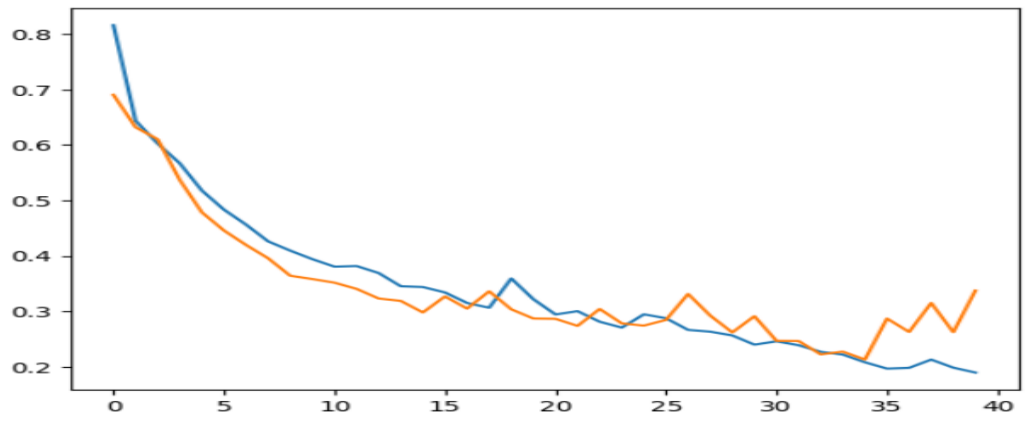
  x = Dense(128, activation='relu', kernel_regularizer=l2(0.001))(x)
  x = Dense(64, activation='relu', kernel_regularizer=l2(0.001))(x)
  x = Dropout(0.25)(x)

1. **Model summary**

```
[ ]  Model: "Cashier_classification"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 this-the_input_layer (Input  [(None, 224, 224, 3)]    0
 Layer)

 layer_1 (Conv2D)            (None, 224, 224, 32)      896

 layer_2 (Conv2D)            (None, 224, 224, 64)      18496

 maxplool1 (MaxPooling2D)    (None, 112, 112, 64)      0

 dropout_2 (Dropout)         (None, 112, 112, 64)      0

 layer_3 (Conv2D)            (None, 112, 112, 32)      18464

 maxplool2 (MaxPooling2D)    (None, 56, 56, 32)        0

 layer_4 (Conv2D)            (None, 56, 56, 32)        9248

 maxplool3 (MaxPooling2D)    (None, 28, 28, 32)        0

 flat (Flatten)              (None, 25088)             0

 dense_4 (Dense)             (None, 128)               3211392

 dense_5 (Dense)             (None, 64)                8256

 dropout_3 (Dropout)         (None, 64)                0

 classification (Dense)      (None, 6)                 390

=================================================================
Total params: 3,267,142
Trainable params: 3,267,142
Non-trainable params: 0
```

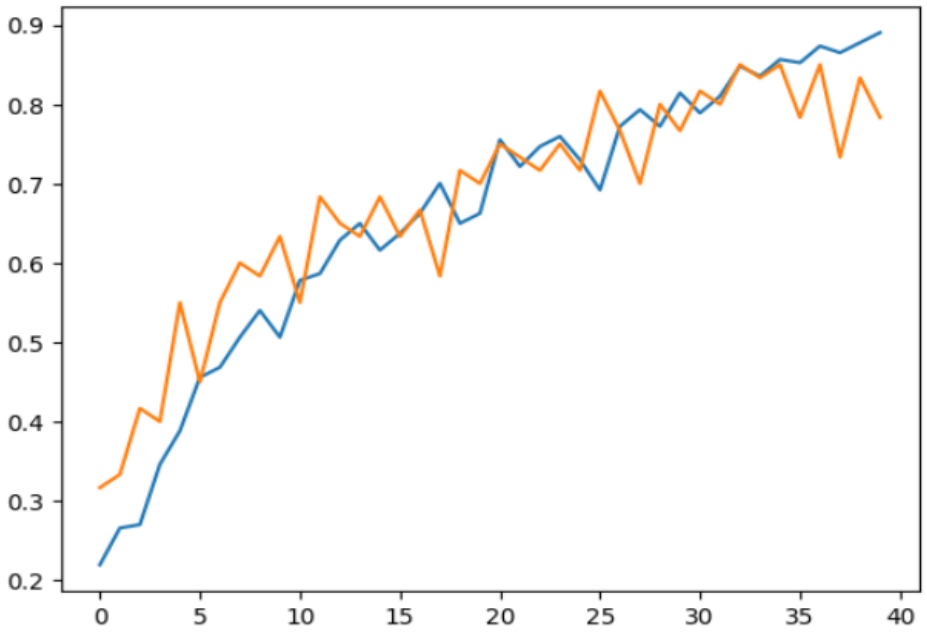
2. Model losses

```
plt.show()
```



3. **Model accuracy**

d



4. **Observation**

   train acc is 0.89 and test acc is 0.78 so over fitting increased and the accuracy decreased

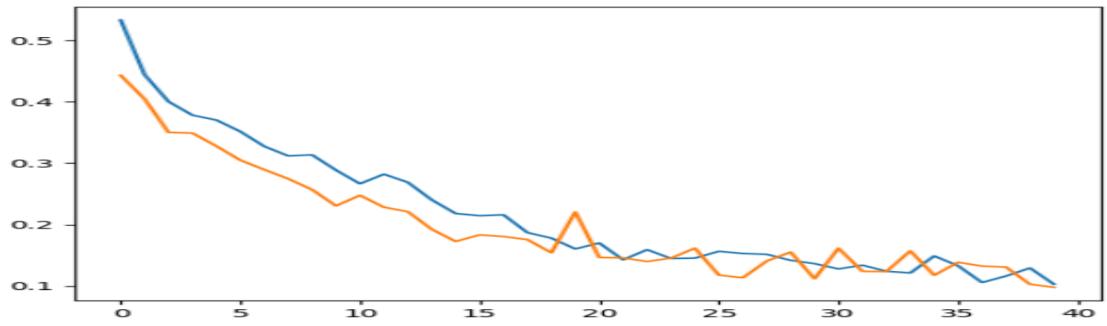
- **Fifith model_4**

  we removed the two regularization layers and one dropout layer and changed the drop out to 20 instead of 25 and increased the data again by augmentation technique

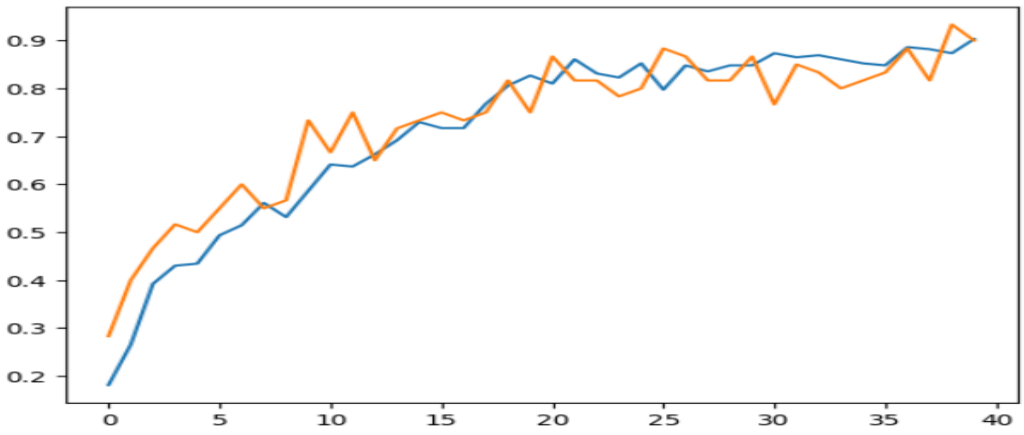
1. **Model summary**

```
Model: "Cashier_classification"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 this-the_input_layer (Input  [(None, 224, 224, 3)]    0
 Layer)

 layer_1 (Conv2D)            (None, 224, 224, 32)      896

 layer_2 (Conv2D)            (None, 224, 224, 64)      18496

 layer_4 (Conv2D)            (None, 224, 224, 128)     73856

 maxplool1 (MaxPooling2D)    (None, 112, 112, 128)     0

 layer_5 (Conv2D)            (None, 112, 112, 32)      36896

 maxplool2 (MaxPooling2D)    (None, 56, 56, 32)        0

 layer_6 (Conv2D)            (None, 56, 56, 32)        9248

 maxplool3 (MaxPooling2D)    (None, 28, 28, 32)        0

 flat (Flatten)              (None, 25088)             0

 classifer (Dense)           (None, 64)                1605696

 dropout_7 (Dropout)         (None, 64)                0

 classification (Dense)      (None, 6)                 390

=================================================================
Total params: 1,745,478
Trainable params: 1,745,478
Non-trainable params: 0
```

2. Model losses



3. **Model accuracy**



4. **Observation**
   train accuracy is 90 and test accuracy is 90 this is the best model fit and best test accuracy we reached

- **Fifth model_4 edited.**
  We increased 20 more epoch to model 40 to solve high bias problem as both the train and test accuracy increase together as the model train longer keeping the model with good fit

Result: train accuracy is 93 and the test accuracy is 90

- **Conclusion:**

This is the table showing the 5 models train and test accuracy.

model	Train acc	Test acc
Model_1	100%	70%
Model_2	96%	90%
Model_3	89%	78%
Model_4	90%	90%
Model_4 edit	93%	93%