```
import tensorflow as tf
from tensorflow.keras import datasets, layers, models
import matplotlib.pyplot as plt
import numpy as np
import os

from keras.models import Sequential
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D, AveragePooling2D
from keras.layers import Flatten,Dense, Dropout

```

```
classes = ['Centerstack','Forward','Left_wing_mirror','Rearview_mirror','Right_wing_mirror
```

```
map_inv = {}

for i in range(len(classes)):
  map_inv[i] = classes[i]

```

```
map_inv
```

```
{0: 'Centerstack',
 1: 'Forward',
 2: 'Left_wing_mirror',
 3: 'Rearview_mirror',
 4: 'Right_wing_mirror'}
```

```
data_dir = '/content/drive/MyDrive/Dataset/gaze_dataset/face'


BATCH_SIZE = 32


train_dataset = tf.keras.utils.image_dataset_from_directory(data_dir,
                                                            shuffle=True,
                                                            batch_size=BATCH_SIZE,
                                                            )


val_ds = tf.keras.utils.image_dataset_from_directory(
  data_dir,
  validation_split=0.2,
  subset="validation",
  seed=123,
  batch_size=8)

```

```

normalization_layer = tf.keras.layers.Rescaling(1./255)
normalized_ds = train_dataset.map(lambda x, y: (normalization_layer(x), y))
image_batch, labels_batch = next(iter(normalized_ds))

AUTOTUNE = tf.data.AUTOTUNE

train_ds = train_dataset.cache().prefetch(buffer_size=AUTOTUNE)
```

```
num_classes = 5

model = tf.keras.Sequential([

  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Conv2D(64, 3, activation='relu'),
  tf.keras.layers.MaxPooling2D(),
  tf.keras.layers.Dropout((0.2)),
  tf.keras.layers.BatchNormalization(),

  tf.keras.layers.Flatten(),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Dense(num_classes,activation = 'softmax')
])

model.compile(
  optimizer='adam',
  loss=tf.losses.SparseCategoricalCrossentropy(from_logits=True),
  metrics=['accuracy'])

model.fit(
  train_ds,
  validation_data=val_ds,
  epochs=25
)
```

```
Found 2415 files belonging to 5 classes.
Found 2415 files belonging to 5 classes.
Using 483 files for validation.
Epoch 1/25
/usr/local/lib/python3.7/dist-packages/tensorflow/python/util/dispatch.py:1082: UserWarn
  return dispatch_target(*args, **kwargs)
76/76 [==============================] - 176s 2s/step - loss: 62.9818 - accuracy: 0.5180
Epoch 2/25
76/76 [==============================] - 16s 210ms/step - loss: 37.4765 - accuracy: 0.69
Epoch 3/25
76/76 [==============================] - 16s 211ms/step - loss: 27.6805 - accuracy: 0.77
Epoch 4/25
76/76 [==============================] - 16s 209ms/step - loss: 28.0516 - accuracy: 0.77
Epoch 5/25
76/76 [==============================] - 16s 210ms/step - loss: 18.4051 - accuracy: 0.83
Epoch 6/25
76/76 [==============================] - 16s 209ms/step - loss: 17.9496 - accuracy: 0.84
```

```
Epoch 7/25
76/76 [==============================] - 16s 209ms/step - loss: 12.5614 - accuracy: 0.89
Epoch 8/25
76/76 [==============================] - 16s 208ms/step - loss: 13.4649 - accuracy: 0.89
Epoch 9/25
76/76 [==============================] - 16s 210ms/step - loss: 9.3308 - accuracy: 0.906
Epoch 10/25
76/76 [==============================] - 16s 210ms/step - loss: 9.0169 - accuracy: 0.919
Epoch 11/25
76/76 [==============================] - 16s 210ms/step - loss: 6.7921 - accuracy: 0.931
Epoch 12/25
76/76 [==============================] - 16s 210ms/step - loss: 6.5698 - accuracy: 0.935
Epoch 13/25
76/76 [==============================] - 16s 209ms/step - loss: 7.8184 - accuracy: 0.933
Epoch 14/25
76/76 [==============================] - 16s 209ms/step - loss: 6.4011 - accuracy: 0.946
Epoch 15/25
76/76 [==============================] - 16s 210ms/step - loss: 5.8920 - accuracy: 0.953
Epoch 16/25
76/76 [==============================] - 16s 210ms/step - loss: 5.1236 - accuracy: 0.958
Epoch 17/25
76/76 [==============================] - 16s 209ms/step - loss: 6.3156 - accuracy: 0.945
Epoch 18/25
76/76 [==============================] - 16s 211ms/step - loss: 4.3435 - accuracy: 0.959
Epoch 19/25
76/76 [==============================] - 16s 210ms/step - loss: 3.1310 - accuracy: 0.968
Epoch 20/25
76/76 [==============================] - 16s 210ms/step - loss: 2.2595 - accuracy: 0.971
Epoch 21/25
76/76 [==============================] - 16s 211ms/step - loss: 2.2658 - accuracy: 0.973
Epoch 22/25
76/76 [==============================] - 16s 211ms/step - loss: 3.4571 - accuracy: 0.965
Epoch 23/25
76/76 [==============================] - 16s 212ms/step - loss: 3.6076 - accuracy: 0.966
Epoch 24/25
76/76 [==============================] - 16s 210ms/step - loss: 3.0483 - accuracy: 0.974
Epoch 25/25
76/76 [==============================] - 16s 211ms/step - loss: 3.9495 - accuracy: 0.966
<keras.callbacks.History at 0x7fe033092f10>
```

```
image_path = "/content/drive/MyDrive/Dataset/gaze_dataset/face_test/1.jpg"
image = tf.keras.preprocessing.image.load_img(image_path,target_size=(256,256))
input_arr = tf.keras.preprocessing.image.img_to_array(image)
input_arr = np.array([input_arr])
predictions = model.predict(input_arr)
```

```
np.argmax(predictions)
```

```
1
```

```
predictions
```

```
array([[0., 1., 0., 0., 0.]], dtype=float32)
```

```
temp_dict = {}
temp_dict['filename'] = []
temp_dict['class'] = []
```

```
t = "/content/drive/MyDrive/Dataset/gaze_dataset/face_test"

for f in os.listdir(t):
  i_path = os.path.join(t,f)

  image_path = i_path
  image = tf.keras.preprocessing.image.load_img(image_path,target_size=(256,256))
  input_arr = tf.keras.preprocessing.image.img_to_array(image)
  input_arr = np.array([input_arr])
  predictions = model.predict(input_arr)

  y = f.split('.')

  x = y[0]+'.'+y[1]
  # print(x)
  temp_dict['filename'].append(x)




  # dicti['class'].append(map_inv[np.argmax(predictions)])

  x = np.argmax(predictions,axis = 1)
  temp_dict['class'].append(map_inv[x[0]])

```

```
import pandas as pd
```

```
df = pd.DataFrame.from_dict(temp_dict)
```

```
df.to_csv("jsr_final.csv", index = False)
```

TRY-2

```
model2 = tf.keras.Sequential([

  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Conv2D(128, 3, activation='relu'),
  tf.keras.layers.MaxPooling2D(),
```

```
  tf.keras.layers.Dropout((0.2)),

  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Conv2D(64, 3, activation='relu'),
  tf.keras.layers.MaxPooling2D(),
  tf.keras.layers.Dropout((0.2)),

  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Flatten(),
  # tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Dense(num_classes,activation = 'softmax')
])

model2.compile(
  optimizer='adam',
  loss=tf.losses.SparseCategoricalCrossentropy(from_logits=True),
  metrics=['accuracy'])

model2.fit(
  train_ds,
  validation_data=val_ds,
  epochs=25
)
```

```
Found 2415 files belonging to 5 classes.
Found 2415 files belonging to 5 classes.
Using 483 files for validation.
Epoch 1/25
/usr/local/lib/python3.7/dist-packages/tensorflow/python/util/dispatch.py:1082: UserWarn
  return dispatch_target(*args, **kwargs)
76/76 [==============================] - 35s 411ms/step - loss: 13.0071 - accuracy: 0.52
Epoch 2/25
76/76 [==============================] - 29s 377ms/step - loss: 8.5950 - accuracy: 0.688
Epoch 3/25
76/76 [==============================] - 29s 377ms/step - loss: 5.6312 - accuracy: 0.774
Epoch 4/25
76/76 [==============================] - 29s 375ms/step - loss: 4.4066 - accuracy: 0.829
Epoch 5/25
76/76 [==============================] - 29s 375ms/step - loss: 2.9663 - accuracy: 0.861
Epoch 6/25
76/76 [==============================] - 28s 374ms/step - loss: 1.6707 - accuracy: 0.914
Epoch 7/25
76/76 [==============================] - 28s 375ms/step - loss: 1.6911 - accuracy: 0.913
Epoch 8/25
76/76 [==============================] - 29s 375ms/step - loss: 1.1804 - accuracy: 0.936
Epoch 9/25
76/76 [==============================] - 28s 374ms/step - loss: 1.0436 - accuracy: 0.940
Epoch 10/25
76/76 [==============================] - 28s 374ms/step - loss: 0.7691 - accuracy: 0.956
Epoch 11/25
76/76 [==============================] - 28s 374ms/step - loss: 0.5179 - accuracy: 0.976
Epoch 12/25
76/76 [==============================] - 29s 375ms/step - loss: 0.8732 - accuracy: 0.963
Epoch 13/25
```

```
76/76 [==============================] - 28s 374ms/step - loss: 0.8108 - accuracy: 0.952
Epoch 14/25
76/76 [==============================] - 28s 375ms/step - loss: 0.8884 - accuracy: 0.954
Epoch 15/25
76/76 [==============================] - 28s 374ms/step - loss: 0.4475 - accuracy: 0.973
Epoch 16/25
76/76 [==============================] - 28s 375ms/step - loss: 0.4716 - accuracy: 0.974
Epoch 17/25
76/76 [==============================] - 28s 374ms/step - loss: 0.4176 - accuracy: 0.975
Epoch 18/25
76/76 [==============================] - 29s 374ms/step - loss: 0.5613 - accuracy: 0.974
Epoch 19/25
76/76 [==============================] - 28s 374ms/step - loss: 0.3660 - accuracy: 0.980
Epoch 20/25
76/76 [==============================] - 29s 375ms/step - loss: 0.4085 - accuracy: 0.978
Epoch 21/25
76/76 [==============================] - 28s 374ms/step - loss: 0.2452 - accuracy: 0.985
Epoch 22/25
76/76 [==============================] - 28s 374ms/step - loss: 0.2716 - accuracy: 0.985
Epoch 23/25
76/76 [==============================] - 28s 374ms/step - loss: 0.1656 - accuracy: 0.988
Epoch 24/25
76/76 [==============================] - 28s 374ms/step - loss: 0.2980 - accuracy: 0.982
Epoch 25/25
76/76 [==============================] - 28s 374ms/step - loss: 0.3670 - accuracy: 0.981
<keras.callbacks.History at 0x7fe031bca410>
```

```
temp_dict = {}
temp_dict['filename'] = []
temp_dict['class'] = []
```

```
t = "/content/drive/MyDrive/Dataset/gaze_dataset/face_test"

for f in os.listdir(t):
  i_path = os.path.join(t,f)

  image_path = i_path
  image = tf.keras.preprocessing.image.load_img(image_path,target_size=(256,256))
  input_arr = tf.keras.preprocessing.image.img_to_array(image)
  input_arr = np.array([input_arr])
  predictions = model2.predict(input_arr)

  y = f.split('.')

  x = y[0]+'.'+y[1]
  # print(x)
  temp_dict['filename'].append(x)
```

```
  # dicti['class'].append(map_inv[np.argmax(predictions)])

  x = np.argmax(predictions,axis = 1)

  temp_dict['class'].append(map_inv[x[0]])
```

```
df = pd.DataFrame.from_dict(temp_dict)
```

```
df.to_csv("jsr_jm_final1.csv", index = False)
```

Try-3

```
from tensorflow.keras.models import Model
from tensorflow.keras.applications.vgg19 import VGG19
from tensorflow.keras.applications.vgg19 import preprocess_input
```

```
vgg = VGG19(input_shape=(256, 256, 3), weights='imagenet', include_top=False)

```

```
for layer in vgg.layers:
    layer.trainable = False
```

```
x = Flatten()(vgg.output)
x = Dropout(0.2)(x)
# x = Dense(8,activation = "relu")(x)
# x = Dropout(0.1)(x)
prediction = Dense(num_classes, activation='softmax')(x)

model3 = Model(inputs=vgg.input, outputs=prediction)
```

```
model3.compile(
  optimizer='adam',
  loss=tf.losses.SparseCategoricalCrossentropy(from_logits=True),
  metrics=['accuracy'])

model3.fit(
  train_ds,
  validation_data=val_ds,
  epochs=15
)
```

```
Epoch 1/15
/usr/local/lib/python3.7/dist-packages/tensorflow/python/util/dispatch.py:1082: UserWarn
  return dispatch_target(*args, **kwargs)
76/76 [==============================] - 42s 543ms/step - loss: 9.1495 - accuracy: 0.632
```

```
Epoch 2/15
76/76 [==============================] - 41s 537ms/step - loss: 2.7067 - accuracy: 0.862
Epoch 3/15
76/76 [==============================] - 41s 540ms/step - loss: 2.1445 - accuracy: 0.905
Epoch 4/15
76/76 [==============================] - 41s 539ms/step - loss: 1.1912 - accuracy: 0.936
Epoch 5/15
76/76 [==============================] - 41s 541ms/step - loss: 0.8311 - accuracy: 0.950
Epoch 6/15
76/76 [==============================] - 41s 537ms/step - loss: 0.8397 - accuracy: 0.955
Epoch 7/15
76/76 [==============================] - 41s 540ms/step - loss: 0.6114 - accuracy: 0.963
Epoch 8/15
76/76 [==============================] - 41s 540ms/step - loss: 0.9555 - accuracy: 0.952
Epoch 9/15
76/76 [==============================] - 41s 540ms/step - loss: 0.9792 - accuracy: 0.959
Epoch 10/15
76/76 [==============================] - 41s 540ms/step - loss: 0.4306 - accuracy: 0.975
Epoch 11/15
76/76 [==============================] - 41s 540ms/step - loss: 0.4705 - accuracy: 0.978
Epoch 12/15
76/76 [==============================] - 41s 541ms/step - loss: 0.4565 - accuracy: 0.976
Epoch 13/15
76/76 [==============================] - 41s 541ms/step - loss: 0.2954 - accuracy: 0.979
Epoch 14/15
76/76 [==============================] - 41s 541ms/step - loss: 0.2598 - accuracy: 0.983
Epoch 15/15
76/76 [==============================] - 41s 538ms/step - loss: 0.0744 - accuracy: 0.991
<keras.callbacks.History at 0x7fdfb7f54850>
```

```
temp_dict = {}
temp_dict['filename'] = []
temp_dict['class'] = []
```

```
t = "/content/drive/MyDrive/Dataset/gaze_dataset/face_test"

for f in os.listdir(t):
  i_path = os.path.join(t,f)

  image_path = i_path
  image = tf.keras.preprocessing.image.load_img(image_path,target_size=(256,256))
  input_arr = tf.keras.preprocessing.image.img_to_array(image)
  input_arr = np.array([input_arr])
  predictions = model3.predict(input_arr)

  y = f.split('.')

  x = y[0]+'.'+y[1]
  # print(x)
  temp_dict['filename'].append(x)

```

```
18
19
20
21   # dicti['class'].append(map_inv[np.argmax(predictions)])
22
23   x = np.argmax(predictions,axis = 1)
24
25   temp_dict['class'].append(map_inv[x[0]])
```

```
1 df = pd.DataFrame.from_dict(temp_dict)
```

```
1 df.to_csv("jsr_jm_final2.csv", index = False)
```

Try-4

```
1 from tensorflow.keras.models import Model
2 from tensorflow.keras.applications.vgg19 import VGG19
3 from tensorflow.keras.applications.vgg19 import preprocess_input
```

```
1 vgg = VGG19(input_shape=(256, 256, 3), weights='imagenet', include_top=False)
```

```
1 for layer in vgg.layers:
2     layer.trainable = False
```

```
1 x = Flatten()(vgg.output)
2 x = Dropout(0.3)(x)
3 # x = Dense(8,activation = "relu")(x)
4 # x = Dropout(0.1)(x)
5 prediction = Dense(num_classes, activation='softmax')(x)
6
7 model4 = Model(inputs=vgg.input, outputs=prediction)
```

```
1 model4.compile(
2   optimizer='adam',
3   loss=tf.losses.SparseCategoricalCrossentropy(from_logits=True),
4   metrics=['accuracy'])
5
6 model4.fit(
7   train_ds,
8   validation_data=val_ds,
9   epochs=16
10 )
```

```
Epoch 1/16
/usr/local/lib/python3.7/dist-packages/tensorflow/python/util/dispatch.py:1082: UserWarn
```

```
  return dispatch_target(*args, **kwargs)
76/76 [==============================] - 42s 538ms/step - loss: 8.9295 - accuracy: 0.621
Epoch 2/16
76/76 [==============================] - 41s 537ms/step - loss: 2.9296 - accuracy: 0.839
Epoch 3/16
76/76 [==============================] - 41s 539ms/step - loss: 2.1188 - accuracy: 0.888
Epoch 4/16
76/76 [==============================] - 41s 540ms/step - loss: 1.7975 - accuracy: 0.909
Epoch 5/16
76/76 [==============================] - 41s 540ms/step - loss: 1.3630 - accuracy: 0.937
Epoch 6/16
76/76 [==============================] - 41s 540ms/step - loss: 1.2138 - accuracy: 0.937
Epoch 7/16
76/76 [==============================] - 41s 540ms/step - loss: 0.9745 - accuracy: 0.947
Epoch 8/16
76/76 [==============================] - 41s 540ms/step - loss: 1.1752 - accuracy: 0.949
Epoch 9/16
76/76 [==============================] - 41s 540ms/step - loss: 0.6665 - accuracy: 0.965
Epoch 10/16
76/76 [==============================] - 41s 540ms/step - loss: 0.5219 - accuracy: 0.971
Epoch 11/16
76/76 [==============================] - 41s 540ms/step - loss: 0.5990 - accuracy: 0.971
Epoch 12/16
76/76 [==============================] - 41s 539ms/step - loss: 0.3866 - accuracy: 0.979
Epoch 13/16
76/76 [==============================] - 41s 541ms/step - loss: 0.3968 - accuracy: 0.982
Epoch 14/16
76/76 [==============================] - 41s 539ms/step - loss: 0.4684 - accuracy: 0.978
Epoch 15/16
76/76 [==============================] - 41s 539ms/step - loss: 0.6778 - accuracy: 0.966
Epoch 16/16
76/76 [==============================] - 41s 540ms/step - loss: 0.7075 - accuracy: 0.967
<keras.callbacks.History at 0x7fdf3d1be0d0>
```

```
temp_dict = {}
temp_dict['filename'] = []
temp_dict['class'] = []
```

```
temp_dict = {}
temp_dict['filename'] = []
temp_dict['class'] = []

t = "/content/drive/MyDrive/Dataset/gaze_dataset/face_test"

for f in os.listdir(t):
  i_path = os.path.join(t,f)

  image_path = i_path
  image = tf.keras.preprocessing.image.load_img(image_path,target_size=(256,256))
  input_arr = tf.keras.preprocessing.image.img_to_array(image)
  input_arr = np.array([input_arr])
  predictions = model4.predict(input_arr)
```

```
  y = f.split('.')

  x = y[0]+'.'+y[1]
  # print(x)
  temp_dict['filename'].append(x)




  # dicti['class'].append(map_inv[np.argmax(predictions)])

  x = np.argmax(predictions,axis = 1)

  temp_dict['class'].append(map_inv[x[0]])
```

```
df = pd.DataFrame.from_dict(temp_dict)
```

```
df.to_csv("jsr_jm_final3.csv", index = False)
```

Try-5

```
resnet_v2 = tf.keras.applications.resnet_v2.ResNet50V2(
    include_top=False,
    weights='imagenet',
    input_tensor=tf.keras.Input(shape=(256, 256, 3)),
)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resne
94674944/94668760 [==============================] - 1s 0us/step
94683136/94668760 [==============================] - 1s 0us/step
```

```
for layer in resnet_v2.layers:
    layer.trainable = False
```

```
x = Flatten()(resnet_v2.output)
# x = Dropout(0.2)(x)
# x = Dense(8,activation = "relu")(x)
# x = Dropout(0.1)(x)
prediction = Dense(num_classes, activation='softmax')(x)

model5 = Model(inputs=resnet_v2.input, outputs=prediction)
```

```
model5.compile(
```

```
  optimizer='adam',
  loss=tf.losses.SparseCategoricalCrossentropy(from_logits=True),
  metrics=['accuracy'])

model5.fit(
  train_ds,
  validation_data=val_ds,
  epochs=15
)
```

```
Epoch 1/15
/usr/local/lib/python3.7/dist-packages/tensorflow/python/util/dispatch.py:1082: UserWarn
  return dispatch_target(*args, **kwargs)
76/76 [==============================] - 35s 392ms/step - loss: 806.2435 - accuracy: 0.3
Epoch 2/15
76/76 [==============================] - 21s 281ms/step - loss: 126.3006 - accuracy: 0.6
Epoch 3/15
76/76 [==============================] - 21s 281ms/step - loss: 161.6694 - accuracy: 0.6
Epoch 4/15
76/76 [==============================] - 21s 282ms/step - loss: 150.9069 - accuracy: 0.6
Epoch 5/15
76/76 [==============================] - 21s 282ms/step - loss: 134.0124 - accuracy: 0.6
Epoch 6/15
76/76 [==============================] - 21s 281ms/step - loss: 60.5521 - accuracy: 0.78
Epoch 7/15
76/76 [==============================] - 21s 281ms/step - loss: 94.6364 - accuracy: 0.74
Epoch 8/15
76/76 [==============================] - 21s 282ms/step - loss: 79.8678 - accuracy: 0.76
Epoch 9/15
76/76 [==============================] - 21s 281ms/step - loss: 98.2920 - accuracy: 0.76
Epoch 10/15
76/76 [==============================] - 21s 281ms/step - loss: 161.1734 - accuracy: 0.7
Epoch 11/15
76/76 [==============================] - 21s 282ms/step - loss: 158.5797 - accuracy: 0.7
Epoch 12/15
76/76 [==============================] - 21s 281ms/step - loss: 180.8649 - accuracy: 0.7
Epoch 13/15
76/76 [==============================] - 21s 280ms/step - loss: 130.1819 - accuracy: 0.7
Epoch 14/15
76/76 [==============================] - 21s 282ms/step - loss: 61.9048 - accuracy: 0.86
Epoch 15/15
76/76 [==============================] - 21s 281ms/step - loss: 56.9795 - accuracy: 0.85
<keras.callbacks.History at 0x7fe0334743d0>
```

Try-6

```
from tensorflow.keras.models import Model
from tensorflow.keras.applications.vgg16 import VGG16
from tensorflow.keras.applications.vgg16 import preprocess_input
```

```
for layer in resnet_v2.layers:
    layer.trainable = False
```

```
x = Flatten()(vgg.output)
x = Dropout(0.2)(x)
# x = Dense(8,activation = "relu")(x)
# x = Dropout(0.1)(x)
prediction = Dense(num_classes, activation='softmax')(x)

model6 = Model(inputs=vgg.input, outputs=prediction)
```

```
model6.compile(
  optimizer='adam',
  loss=tf.losses.SparseCategoricalCrossentropy(from_logits=True),
  metrics=['accuracy'])

model6.fit(
  train_ds,
  validation_data=val_ds,
  epochs=15
)
```

```
Epoch 1/15
/usr/local/lib/python3.7/dist-packages/tensorflow/python/util/dispatch.py:1082: UserWarn
  return dispatch_target(*args, **kwargs)
76/76 [==============================] - 42s 542ms/step - loss: 9.3720 - accuracy: 0.629
Epoch 2/15
76/76 [==============================] - 41s 540ms/step - loss: 2.7251 - accuracy: 0.851
Epoch 3/15
76/76 [==============================] - 41s 540ms/step - loss: 2.0511 - accuracy: 0.898
Epoch 4/15
74/76 [============================>.] - ETA: 0s - loss: 1.4851 - accuracy: 0.9253
```

```
temp_dict = {}
temp_dict['filename'] = []
temp_dict['class'] = []

t = "/content/drive/MyDrive/Dataset/gaze_dataset/face_test"

for f in os.listdir(t):
  i_path = os.path.join(t,f)

  image_path = i_path
  image = tf.keras.preprocessing.image.load_img(image_path,target_size=(256,256))
  input_arr = tf.keras.preprocessing.image.img_to_array(image)
  input_arr = np.array([input_arr])
  predictions = model6.predict(input_arr)

  y = f.split('_')
```

```
y = f.split('.')

x = y[0]+'.'+y[1]
# print(x)
temp_dict['filename'].append(x)




# dicti['class'].append(map_inv[np.argmax(predictions)])

x = np.argmax(predictions,axis = 1)

temp_dict['class'].append(map_inv[x[0]])
```

```
df = pd.DataFrame.from_dict(temp_dict)
```

```
df.to_csv("jsr_jm_final6.csv", index = False)
```