

21BDS0340

Abhinav Dinesh Srivatsa

Deep Learning Lab

#### Assignment – IV

##### **Procedure:**

1. Import the required packages
2. Load the dataset locally
3. Normalize the train features
4. Reshape the train features for the convolutional layers
5. Split the training data into train and validation datasets
6. Plot a random training data sample
7. Build the TensorFlow model
8. Evaluate the model with the test datasets
9. Plot the model accuracy vs validation accuracy
10. Plot the model loss vs validation loss
11. Find the confusion matrix between real and predicted classes
12. Find the F1, precision, recall and support scores for the confusion matrix

**Interactive Python Notebook on the following pages:**

## 21BDS0340 - Abhinav Dinesh Srivatsa

```
In [ ]: import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
import random
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
import seaborn as sns
```

```
In [ ]: data = np.load("ORL_faces.npz")
X_train = data["trainX"]
X_train_norm = np.array(X_train, dtype="float") / 255
X_test = data["testX"]
X_test_norm = np.array(X_test, dtype="float") / 255
y_train = data["trainY"]
y_test = data["testY"]
X_train_norm.shape, y_train.shape
```

```
Out[ ]: ((240, 10304), (240,))
```

```
In [ ]: rows, cols = 112, 92
im_shape = (-1, rows, cols)

X_train_reshaped = np.reshape(X_train_norm, im_shape)
X_test_reshaped = np.reshape(X_test_norm, im_shape)
X_train_reshaped.shape
```

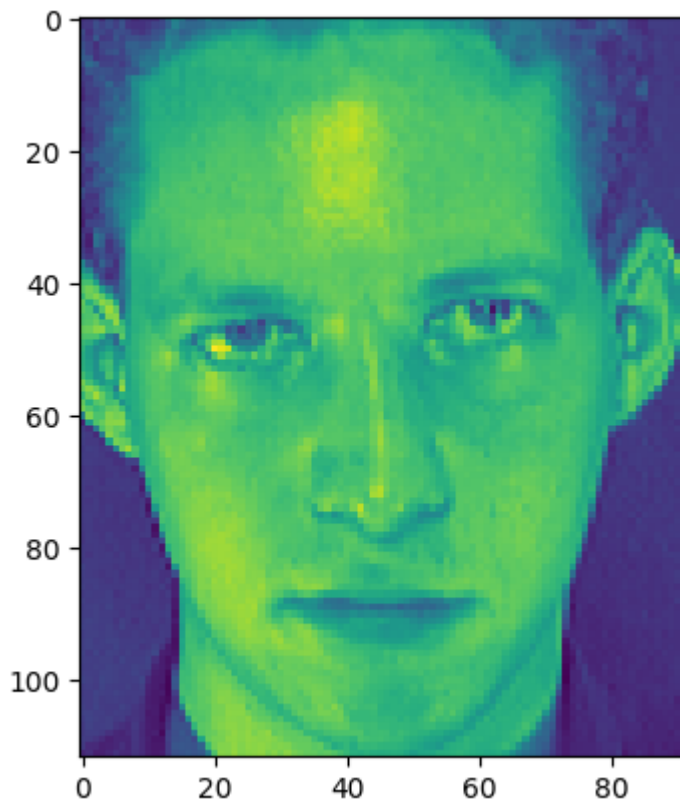
```
Out[ ]: (240, 112, 92)
```

```
In [ ]: X_train, X_val, y_train, y_val = train_test_split(X_train_reshaped, y_train,
X_train.shape, X_val.shape)
```

```
Out[ ]: ((192, 112, 92), (48, 112, 92))
```

```
In [ ]: def random_face(data):
    n = int(random.random() * len(data))
    plt.imshow(data[n])
```


```
In [ ]: random_face(X_train)
```





```
In [ ]: model1 = tf.keras.Sequential([
    tf.keras.layers.Input((rows, cols, 1)),
    tf.keras.layers.Conv2D(36, 7, activation="relu"),
    tf.keras.layers.MaxPool2D(2),
    tf.keras.layers.Conv2D(54, 5, activation="relu"),
    tf.keras.layers.MaxPool2D(2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(2048, activation="relu"),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(1024, activation="relu"),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(512, activation="relu"),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(20, activation="softmax")
])


model1.compile(
    optimizer="adam",
    loss="sparse_categorical_crossentropy",
    metrics=["accuracy"]
)


history1 = model1.fit(
    X_train, y_train,
    epochs=100, validation_data=(X_val, y_val)
)
```


Epoch 1/100  
6/6  24s 2s/step - accuracy: 0.0458 - loss: 3.4466 - val\_accuracy: 0.0417 - val\_loss: 2.9964


Epoch 2/100  
6/6  17s 2s/step - accuracy: 0.0667 - loss: 2.9951 - val\_accuracy: 0.0625 - val\_loss: 2.9951


Epoch 3/100  
6/6  12s 2s/step - accuracy: 0.0689 - loss: 2.9872 - val\_accuracy: 0.0417 - val\_loss: 3.0243


Epoch 4/100  
6/6  20s 2s/step - accuracy: 0.0646 - loss: 2.9982 - val\_accuracy: 0.0625 - val\_loss: 2.9967


Epoch 5/100  
6/6  20s 2s/step - accuracy: 0.1097 - loss: 2.9843 - val\_accuracy: 0.1042 - val\_loss: 2.9938


Epoch 6/100  
6/6  11s 2s/step - accuracy: 0.0850 - loss: 2.9790 - val\_accuracy: 0.0208 - val\_loss: 3.0041


Epoch 7/100  
6/6  19s 2s/step - accuracy: 0.1057 - loss: 2.9901 - val\_accuracy: 0.0417 - val\_loss: 3.0058


Epoch 8/100  
6/6  28s 3s/step - accuracy: 0.0893 - loss: 2.9833 - val\_accuracy: 0.0417 - val\_loss: 3.0322


Epoch 9/100  
6/6  13s 2s/step - accuracy: 0.0936 - loss: 2.9444 - val\_accuracy: 0.1458 - val\_loss: 2.9575


Epoch 10/100  
6/6  21s 2s/step - accuracy: 0.1559 - loss: 2.8848 - val\_accuracy: 0.1875 - val\_loss: 2.8893


Epoch 11/100  
6/6  21s 2s/step - accuracy: 0.2330 - loss: 2.7857 - val\_accuracy: 0.1667 - val\_loss: 2.6574


Epoch 12/100  
6/6  21s 2s/step - accuracy: 0.2153 - loss: 2.5889 - val\_accuracy: 0.3125 - val\_loss: 2.3189


Epoch 13/100  
6/6  20s 2s/step - accuracy: 0.3759 - loss: 2.1222 - val\_accuracy: 0.5000 - val\_loss: 1.9070


Epoch 14/100  
6/6  20s 2s/step - accuracy: 0.4125 - loss: 1.8827 - val\_accuracy: 0.6250 - val\_loss: 1.4847


Epoch 15/100  
6/6  20s 2s/step - accuracy: 0.5778 - loss: 1.3281 - val\_accuracy: 0.7083 - val\_loss: 1.1688


Epoch 16/100  
6/6  21s 2s/step - accuracy: 0.6059 - loss: 1.2618 - val\_accuracy: 0.7083 - val\_loss: 0.9985


Epoch 17/100  
6/6  20s 2s/step - accuracy: 0.6760 - loss: 0.9188 - val\_accuracy: 0.7917 - val\_loss: 0.9341


Epoch 18/100  
6/6  20s 2s/step - accuracy: 0.7065 - loss: 0.8871 - val\_accuracy: 0.8125 - val\_loss: 0.7149


Epoch 19/100  
6/6  21s 2s/step - accuracy: 0.7281 - loss: 0.8559 - val\_accuracy: 0.8333 - val\_loss: 0.4649


Epoch 20/100  
6/6  20s 2s/step - accuracy: 0.8927 - loss: 0.3756 - val\_accuracy: 0.8958 - val\_loss: 0.4468


Epoch 21/100  
6/6  20s 2s/step - accuracy: 0.8911 - loss: 0.3815 - val\_accuracy: 0.8333 - val\_loss: 0.3917


Epoch 22/100  
6/6  20s 2s/step - accuracy: 0.8538 - loss: 0.5548 - val\_accuracy: 0.8958 - val\_loss: 0.2854


Epoch 23/100  
6/6  21s 2s/step - accuracy: 0.9289 - loss: 0.2185 - val\_accuracy: 0.8958 - val\_loss: 0.2595


Epoch 24/100  
6/6  19s 2s/step - accuracy: 0.9275 - loss: 0.1774 - val\_accuracy: 0.8958 - val\_loss: 0.3016


Epoch 25/100  
6/6  22s 2s/step - accuracy: 0.9647 - loss: 0.1512 - val\_accuracy: 0.9375 - val\_loss: 0.2089


Epoch 26/100  
6/6  13s 2s/step - accuracy: 0.9795 - loss: 0.0797 - val\_accuracy: 0.9583 - val\_loss: 0.1741


Epoch 27/100  
6/6  12s 2s/step - accuracy: 0.9652 - loss: 0.1725 - val\_accuracy: 0.8958 - val\_loss: 0.3344


Epoch 28/100  
6/6  19s 2s/step - accuracy: 0.9608 - loss: 0.1135 - val\_accuracy: 0.9375 - val\_loss: 0.2157


Epoch 29/100  
6/6  12s 2s/step - accuracy: 0.9577 - loss: 0.0957 - val\_accuracy: 0.9167 - val\_loss: 0.2877


Epoch 30/100  
6/6  20s 2s/step - accuracy: 0.9891 - loss: 0.0589 - val\_accuracy: 0.9167 - val\_loss: 0.3054


Epoch 31/100  
6/6  20s 2s/step - accuracy: 0.9862 - loss: 0.0641 - val\_accuracy: 0.9375 - val\_loss: 0.2596


Epoch 32/100  
6/6  21s 2s/step - accuracy: 0.9746 - loss: 0.1114 - val\_accuracy: 0.9583 - val\_loss: 0.2686


Epoch 33/100  
6/6  19s 2s/step - accuracy: 0.9920 - loss: 0.0293 - val\_accuracy: 0.9583 - val\_loss: 0.2237


Epoch 34/100  
6/6  11s 2s/step - accuracy: 0.9928 - loss: 0.0566 - val\_accuracy: 0.9583 - val\_loss: 0.1840


Epoch 35/100  
6/6  12s 2s/step - accuracy: 0.9965 - loss: 0.0205 - val\_accuracy: 0.9583 - val\_loss: 0.1724


Epoch 36/100  
6/6  20s 2s/step - accuracy: 1.0000 - loss: 0.0130 - val\_accuracy: 0.9583 - val\_loss: 0.1835


Epoch 37/100  
6/6  21s 2s/step - accuracy: 0.9985 - loss: 0.0105 - val\_accuracy: 0.9583 - val\_loss: 0.1737


Epoch 38/100  
6/6  21s 2s/step - accuracy: 0.9928 - loss: 0.0261 - val\_accuracy: 0.9583 - val\_loss: 0.1658


Epoch 39/100  
6/6  20s 2s/step - accuracy: 0.9885 - loss: 0.0452 - val\_accuracy: 0.9583 - val\_loss: 0.1714


Epoch 40/100  
6/6  26s 3s/step - accuracy: 0.9760 - loss: 0.0386 - val\_accuracy: 0.9375 - val\_loss: 0.2329


Epoch 41/100  
6/6  12s 2s/step - accuracy: 0.9965 - loss: 0.0450 - val\_accuracy: 0.9375 - val\_loss: 0.3381


Epoch 42/100  
6/6  21s 2s/step - accuracy: 0.9890 - loss: 0.0680 - val\_accuracy: 0.9583 - val\_loss: 0.1825


Epoch 43/100  
6/6  20s 2s/step - accuracy: 0.9550 - loss: 0.0801 - val\_accuracy: 0.9375 - val\_loss: 0.2802


Epoch 44/100  
6/6  20s 2s/step - accuracy: 0.9769 - loss: 0.0933 - val\_accuracy: 0.9583 - val\_loss: 0.2271


Epoch 45/100  
6/6  21s 2s/step - accuracy: 0.9760 - loss: 0.0793 - val\_accuracy: 0.9583 - val\_loss: 0.1735


Epoch 46/100  
6/6  20s 2s/step - accuracy: 0.9961 - loss: 0.0187 - val\_accuracy: 0.9583 - val\_loss: 0.2003


Epoch 47/100  
6/6  21s 2s/step - accuracy: 1.0000 - loss: 0.0109 - val\_accuracy: 0.9375 - val\_loss: 0.3036


Epoch 48/100  
6/6  20s 2s/step - accuracy: 0.9928 - loss: 0.0326 - val\_accuracy: 0.9583 - val\_loss: 0.2853


Epoch 49/100  
6/6  21s 2s/step - accuracy: 0.9946 - loss: 0.0114 - val\_accuracy: 0.9583 - val\_loss: 0.2951


Epoch 50/100  
6/6  20s 2s/step - accuracy: 0.9833 - loss: 0.0484 - val\_accuracy: 0.9583 - val\_loss: 0.2710


Epoch 51/100  
6/6  21s 2s/step - accuracy: 0.9883 - loss: 0.0400 - val\_accuracy: 0.9583 - val\_loss: 0.2419


Epoch 52/100  
6/6  12s 2s/step - accuracy: 0.9883 - loss: 0.0321 - val\_accuracy: 0.9583 - val\_loss: 0.2050


Epoch 53/100  
6/6  12s 2s/step - accuracy: 1.0000 - loss: 0.0129 - val\_accuracy: 0.9583 - val\_loss: 0.1982


Epoch 54/100  
6/6  21s 2s/step - accuracy: 0.9965 - loss: 0.0158 - val\_accuracy: 0.9375 - val\_loss: 0.1875


Epoch 55/100  
6/6  12s 2s/step - accuracy: 0.9950 - loss: 0.0140 - val\_accuracy: 0.9167 - val\_loss: 0.2611


Epoch 56/100  
6/6  20s 2s/step - accuracy: 1.0000 - loss: 0.0170 - val\_accuracy: 0.9375 - val\_loss: 0.1979


Epoch 57/100  
6/6  21s 2s/step - accuracy: 1.0000 - loss: 0.0134 - val\_accuracy: 0.9583 - val\_loss: 0.2029


Epoch 58/100  
6/6  20s 2s/step - accuracy: 0.9965 - loss: 0.0108 - val\_accuracy: 0.9583 - val\_loss: 0.2102


Epoch 59/100  
6/6  21s 2s/step - accuracy: 1.0000 - loss: 0.0068 - val\_accuracy: 0.9583 - val\_loss: 0.2105


Epoch 60/100  
6/6  20s 2s/step - accuracy: 0.9833 - loss: 0.0261 - val\_accuracy: 0.9375 - val\_loss: 0.2277


Epoch 61/100  
6/6  21s 2s/step - accuracy: 0.9985 - loss: 0.0072 - val\_accuracy: 0.9167 - val\_loss: 0.2894


Epoch 62/100  
6/6  20s 2s/step - accuracy: 0.9868 - loss: 0.0238 - val\_accuracy: 0.9583 - val\_loss: 0.2485


Epoch 63/100  
6/6  20s 2s/step - accuracy: 1.0000 - loss: 0.0115 - val\_accuracy: 0.9583 - val\_loss: 0.2337


Epoch 64/100  
6/6  22s 2s/step - accuracy: 0.9928 - loss: 0.0239 - val\_accuracy: 0.9583 - val\_loss: 0.2140


Epoch 65/100  
6/6  19s 2s/step - accuracy: 0.9950 - loss: 0.0113 - val\_accuracy: 0.9583 - val\_loss: 0.2145


Epoch 66/100  
6/6  11s 2s/step - accuracy: 0.9859 - loss: 0.0279 - val\_accuracy: 0.9583 - val\_loss: 0.2225


Epoch 67/100  
6/6  21s 2s/step - accuracy: 0.9833 - loss: 0.0266 - val\_accuracy: 0.9583 - val\_loss: 0.2374


Epoch 68/100  
6/6  19s 2s/step - accuracy: 0.9913 - loss: 0.0136 - val\_accuracy: 0.9583 - val\_loss: 0.2635


Epoch 69/100  
6/6  12s 2s/step - accuracy: 1.0000 - loss: 0.0059 - val\_accuracy: 0.9583 - val\_loss: 0.2366


Epoch 70/100  
6/6  12s 2s/step - accuracy: 0.9950 - loss: 0.0376 - val\_accuracy: 0.9583 - val\_loss: 0.2169


Epoch 71/100  
6/6  20s 2s/step - accuracy: 0.9985 - loss: 0.0144 - val\_accuracy: 0.9375 - val\_loss: 0.2309


Epoch 72/100  
6/6  21s 2s/step - accuracy: 1.0000 - loss: 0.0070 - val\_accuracy: 0.9583 - val\_loss: 0.2476


Epoch 73/100  
6/6  28s 3s/step - accuracy: 0.9985 - loss: 0.0039 - val\_accuracy: 0.9583 - val\_loss: 0.2655


Epoch 74/100  
6/6  14s 2s/step - accuracy: 1.0000 - loss: 0.0044 - val\_accuracy: 0.9375 - val\_loss: 0.3651


Epoch 75/100  
6/6  20s 2s/step - accuracy: 0.9976 - loss: 0.0227 - val\_accuracy: 0.9375 - val\_loss: 0.3464


Epoch 76/100  
6/6  20s 2s/step - accuracy: 0.9976 - loss: 0.0122 - val\_accuracy: 0.9583 - val\_loss: 0.3123


Epoch 77/100  
6/6  21s 2s/step - accuracy: 0.9965 - loss: 0.0100 - val\_accuracy: 0.9583 - val\_loss: 0.3317


Epoch 78/100  
6/6  20s 2s/step - accuracy: 1.0000 - loss: 8.2271e-04 - val\_accuracy: 0.9375 - val\_loss: 0.3651


Epoch 79/100  
6/6  20s 2s/step - accuracy: 1.0000 - loss: 0.0029 - val\_accuracy: 0.9375 - val\_loss: 0.2213


Epoch 80/100  
6/6  21s 2s/step - accuracy: 0.9883 - loss: 0.0421 - val\_accuracy: 0.9583 - val\_loss: 0.2712


Epoch 81/100  
6/6  12s 2s/step - accuracy: 0.9965 - loss: 0.0206 - val\_accuracy: 0.9583 - val\_loss: 0.2103


Epoch 82/100  
6/6  12s 2s/step - accuracy: 1.0000 - loss: 0.0225 - val\_accuracy: 0.9583 - val\_loss: 0.1582


Epoch 83/100  
6/6  19s 2s/step - accuracy: 1.0000 - loss: 0.0050 - val\_accuracy: 0.9583 - val\_loss: 0.1884


Epoch 84/100  
6/6  22s 2s/step - accuracy: 0.9915 - loss: 0.0075 - val\_accuracy: 0.9583 - val\_loss: 0.2948


Epoch 85/100  
6/6  12s 2s/step - accuracy: 1.0000 - loss: 0.0076 - val\_accuracy: 0.9583 - val\_loss: 0.3382


Epoch 86/100  
6/6  20s 2s/step - accuracy: 0.9883 - loss: 0.0215 - val\_accuracy: 0.9583 - val\_loss: 0.2866


Epoch 87/100  
6/6  21s 2s/step - accuracy: 0.9935 - loss: 0.0225 - val\_accuracy: 0.9583 - val\_loss: 0.2531


Epoch 88/100  
6/6  19s 2s/step - accuracy: 1.0000 - loss: 0.0076 - val\_accuracy: 0.9583 - val\_loss: 0.2745


Epoch 89/100  
6/6  12s 2s/step - accuracy: 1.0000 - loss: 0.0011 - val\_accuracy: 0.9583 - val\_loss: 0.2697


Epoch 90/100  
6/6  20s 2s/step - accuracy: 1.0000 - loss: 4.7826e-04 - val\_accuracy: 0.9583 - val\_loss: 0.2513


Epoch 91/100  
6/6  21s 2s/step - accuracy: 1.0000 - loss: 0.0034 - val\_accuracy: 0.9583 - val\_loss: 0.2833


Epoch 92/100  
6/6  20s 2s/step - accuracy: 1.0000 - loss: 0.0036 - val\_accuracy: 0.9583 - val\_loss: 0.3397


Epoch 93/100  
6/6  11s 2s/step - accuracy: 0.9976 - loss: 0.0052 - val\_accuracy: 0.9583 - val\_loss: 0.3481


Epoch 94/100  
6/6  11s 2s/step - accuracy: 0.9883 - loss: 0.0186 - val\_accuracy: 0.9375 - val\_loss: 0.4990


Epoch 95/100  
6/6  21s 2s/step - accuracy: 0.9915 - loss: 0.0183 - val\_accuracy: 0.9583 - val\_loss: 0.4127

Epoch 96/100  
6/6  19s 2s/step - accuracy: 0.9883 - loss: 0.0159 - val\_accuracy: 0.9583 - val\_loss: 0.3377

Epoch 97/100  
6/6  22s 2s/step - accuracy: 0.9883 - loss: 0.0414 - val\_accuracy: 0.9167 - val\_loss: 0.3823

Epoch 98/100  
6/6  20s 2s/step - accuracy: 0.9928 - loss: 0.0362 - val\_accuracy: 0.9375 - val\_loss: 0.4841

Epoch 99/100  
6/6  20s 2s/step - accuracy: 1.0000 - loss: 0.0085 - val\_accuracy: 0.8958 - val\_loss: 0.5321

Epoch 100/100  
6/6  21s 2s/step - accuracy: 0.9802 - loss: 0.0418 - val\_accuracy: 0.9583 - val\_loss: 0.2916



```
In [ ]: X_test_resaped.shape
```

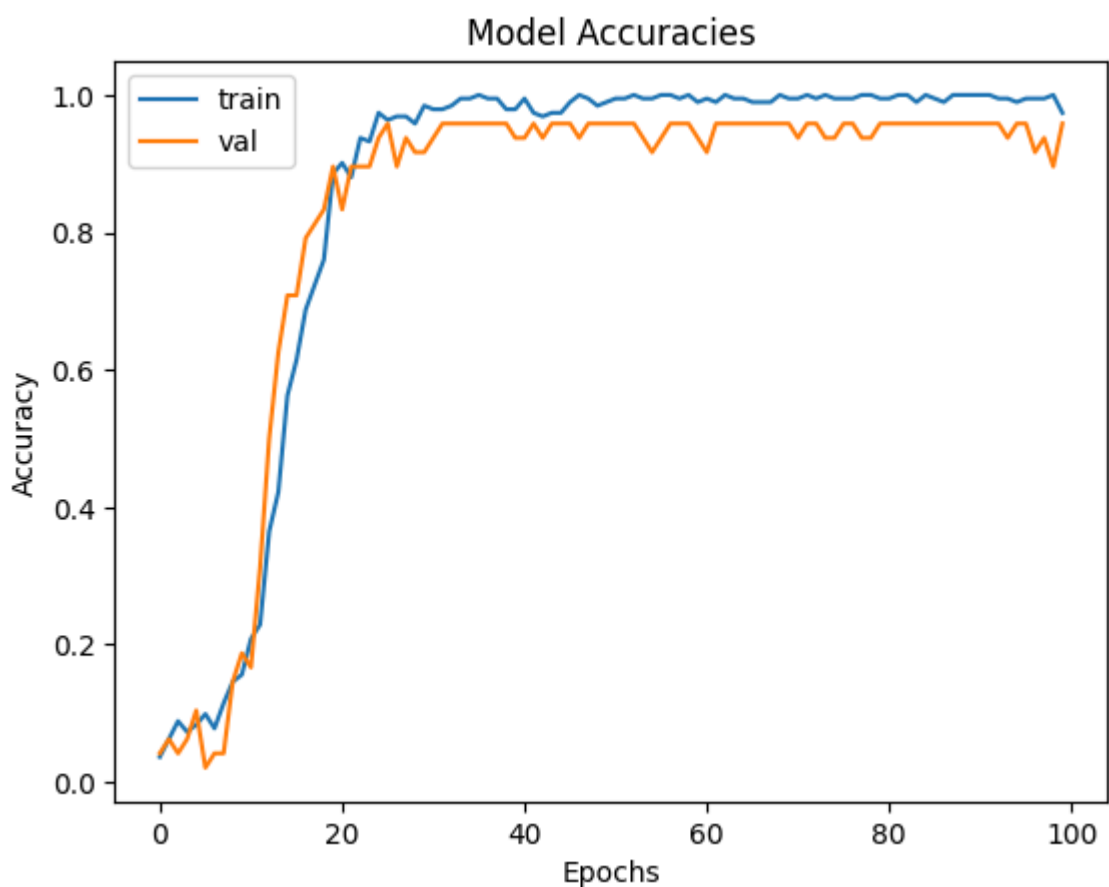
```
Out[ ]: (160, 112, 92)
```

```
In [ ]: model1.evaluate(X_test_resaped, y_test, verbose=0)
```

```
Out[ ]: [1.2689409255981445, 0.8374999761581421]
```

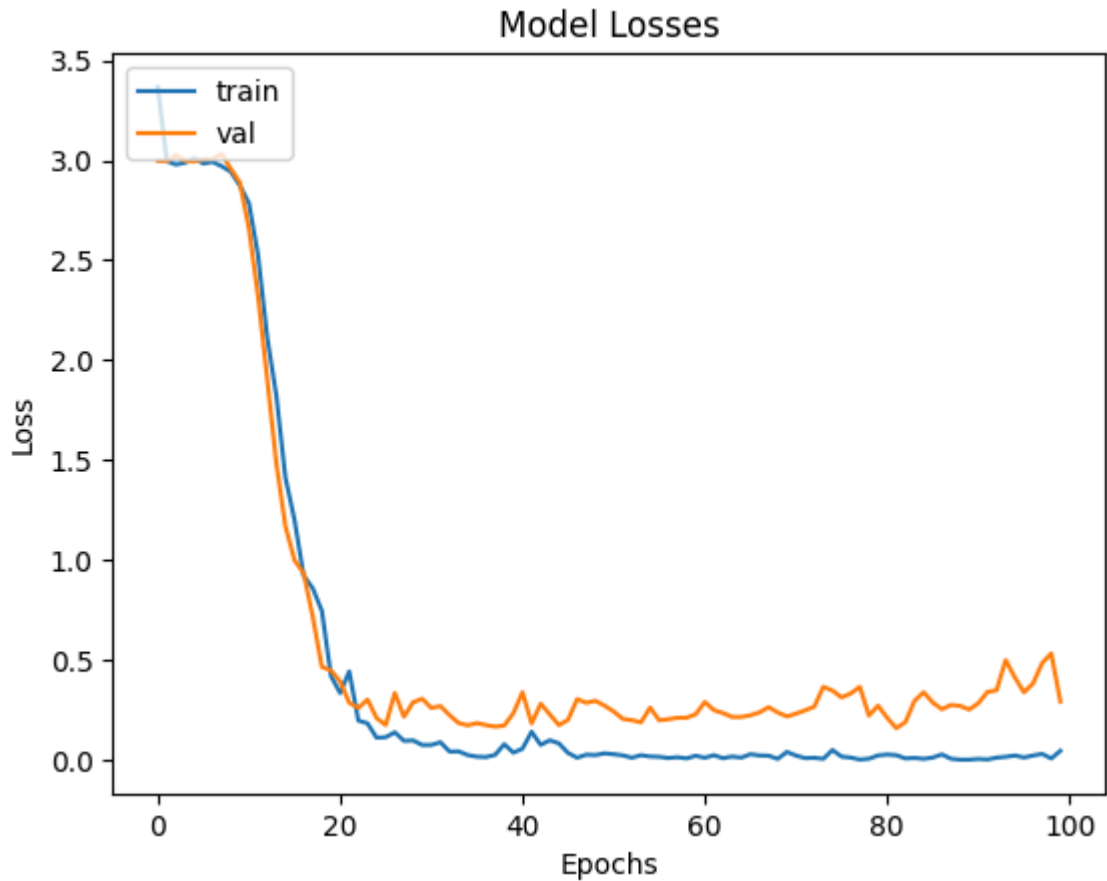
```
In [ ]: plt.plot(history1.history["accuracy"])
plt.plot(history1.history["val_accuracy"])
plt.title("Model Accuracies")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend(["train", "val"], loc="upper left")
```

```
Out[ ]: <matplotlib.legend.Legend at 0x7af7b959f2b0>
```



```
In [ ]: plt.plot(history1.history["loss"])
plt.plot(history1.history["val_loss"])
plt.title("Model Losses")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend(["train", "val"], loc="upper left")
```

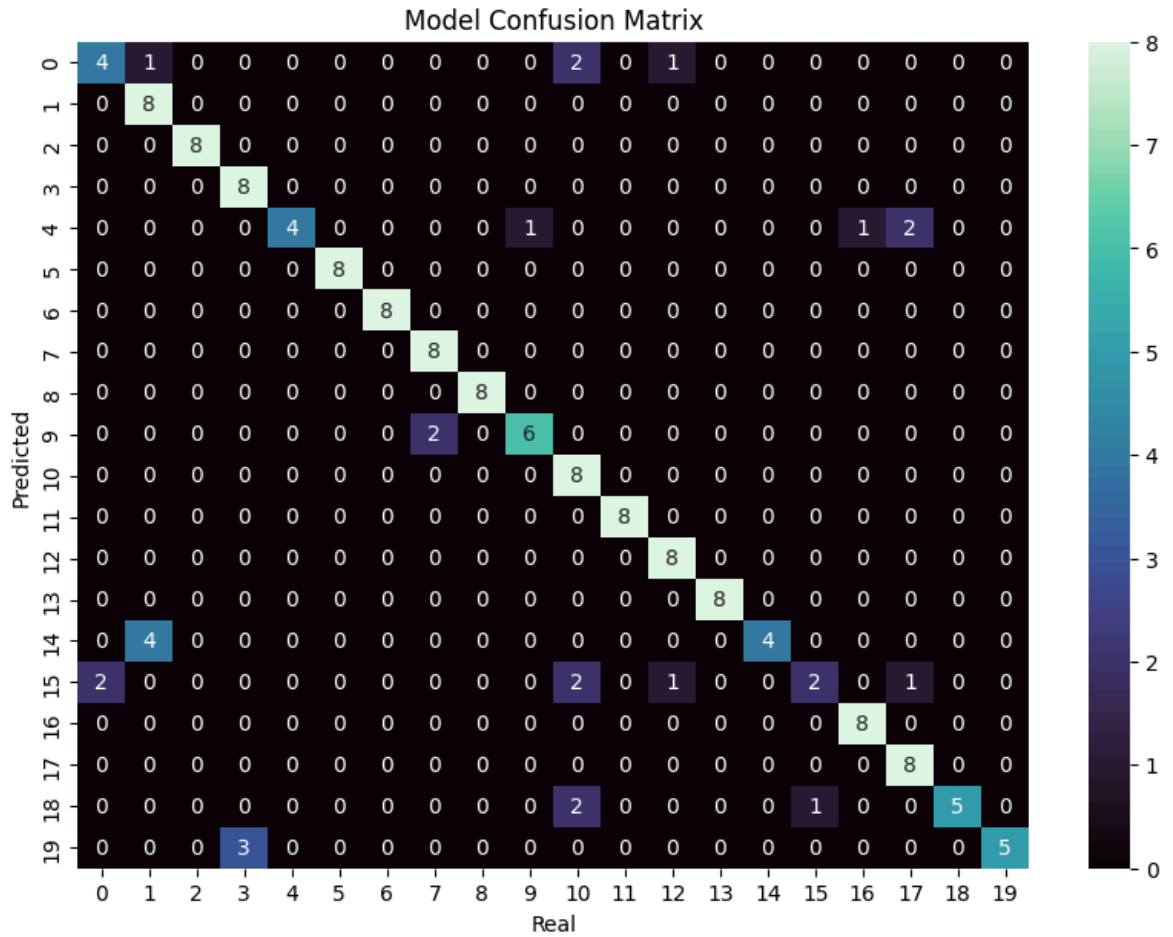
```
Out[ ]: <matplotlib.legend.Legend at 0x7af7b82dbd00>
```



```
In [ ]: def confusion_matrix(model, test_features, test_labels):
    y_pred = model.predict(test_features)
    y_pred = [np.argmax(row) for row in y_pred]
    matrix = np.zeros((20, 20))
    for i in range(len(y_pred)):
        real = test_labels[i]
        pred = y_pred[i]
        matrix[real][pred] += 1
    plt.figure(figsize=(10, 7))
    sns.heatmap(matrix, cmap="mako", annot=True)
    plt.title("Model Confusion Matrix")
    plt.xlabel("Real")
    plt.ylabel("Predicted")
    return y_pred
```

```
In [ ]: y_pred = confusion_matrix(model1, X_test_resaped, y_test)
```

5/5 ————— 3s 533ms/step



```
In [ ]: report = classification_report(y_test, y_pred)
print(report)
```

	precision	recall	f1-score	support
0	0.67	0.50	0.57	8
1	0.62	1.00	0.76	8
2	1.00	1.00	1.00	8
3	0.73	1.00	0.84	8
4	1.00	0.50	0.67	8
5	1.00	1.00	1.00	8
6	1.00	1.00	1.00	8
7	0.80	1.00	0.89	8
8	1.00	1.00	1.00	8
9	0.86	0.75	0.80	8
10	0.57	1.00	0.73	8
11	1.00	1.00	1.00	8
12	0.80	1.00	0.89	8
13	1.00	1.00	1.00	8
14	1.00	0.50	0.67	8
15	0.67	0.25	0.36	8
16	0.89	1.00	0.94	8
17	0.73	1.00	0.84	8
18	1.00	0.62	0.77	8
19	1.00	0.62	0.77	8
accuracy			0.84	160
macro avg	0.87	0.84	0.82	160
weighted avg	0.87	0.84	0.82	160