```python
model_dict = pickle.load(open('./model1.p', 'rb'))
model = model_dict['model']

cap = cv2.VideoCapture(0)

mp_hands = mp.solutions.hands
mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles

hands = mp_hands.Hands(static_image_mode=True, min_detection_confidence=0.5)

labels_dict = {
    0: '0', 1: '1', 2: '2', 3: '3', 4: '4', 5: '5', 6: '6', 7: '7', 8: '8', 9: '9',
    10: 'A', 11: 'B', 12: 'C', 13: 'D', 14: 'E', 15: 'F', 16: 'G', 17: 'H', 18: 'I',
    19: 'J', 20: 'K', 21: 'L', 22: 'M', 23: 'N', 24: 'O', 25: 'P', 26: 'Q', 27: 'R',
    28: 'S', 29: 'T', 30: 'U', 31: 'V', 32: 'W', 33: 'X', 34: 'Y', 35: 'Z', 36: 'Cool'
}
```

```
    # Pad the data_aux list to the fixed length
    padded_data = pad_sequences([data_aux], maxlen=max_length, padding='post', dtype='float32', truncating='post')
    prediction = model.predict(padded_data)
    predicted_label_index = int(prediction[0])

    if predicted_label_index in labels_dict:
        predicted_character = labels_dict[predicted_label_index]
    else:
        predicted_character = ''  # Leave it blank if the label doesn't exist

    x1 = int(min(x_) * W) - 15
    y1 = int(min(y_) * H) - 15
    x2 = int(max(x_) * W) - 15
    y2 = int(max(y_) * H) - 15

    cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 0), 4)
    cv2.putText(frame, predicted_character, (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 1.3, (0, 0, 0), 3,
                cv2.LINE_AA)
    cv2.imshow('frame', frame)
    cv2.waitKey(1)
    print(predicted_character)

key = cv2.waitKey(1)
if key == ord('q'):
    break
```

```
data = np.asarray(padded_data)
labels = np.asarray(data_dict['labels'])
x_train, x_test, y_train, y_test = train_test_split(data, labels, test_size=0.2,
                                                    shuffle=True, stratify=labels,random_state=25)
```

```
model = RandomForestClassifier()
epochs = 20
for epoch in range(epochs):
    model.fit(x_train, y_train)
```

```
y_predict = model.predict(x_test)
```

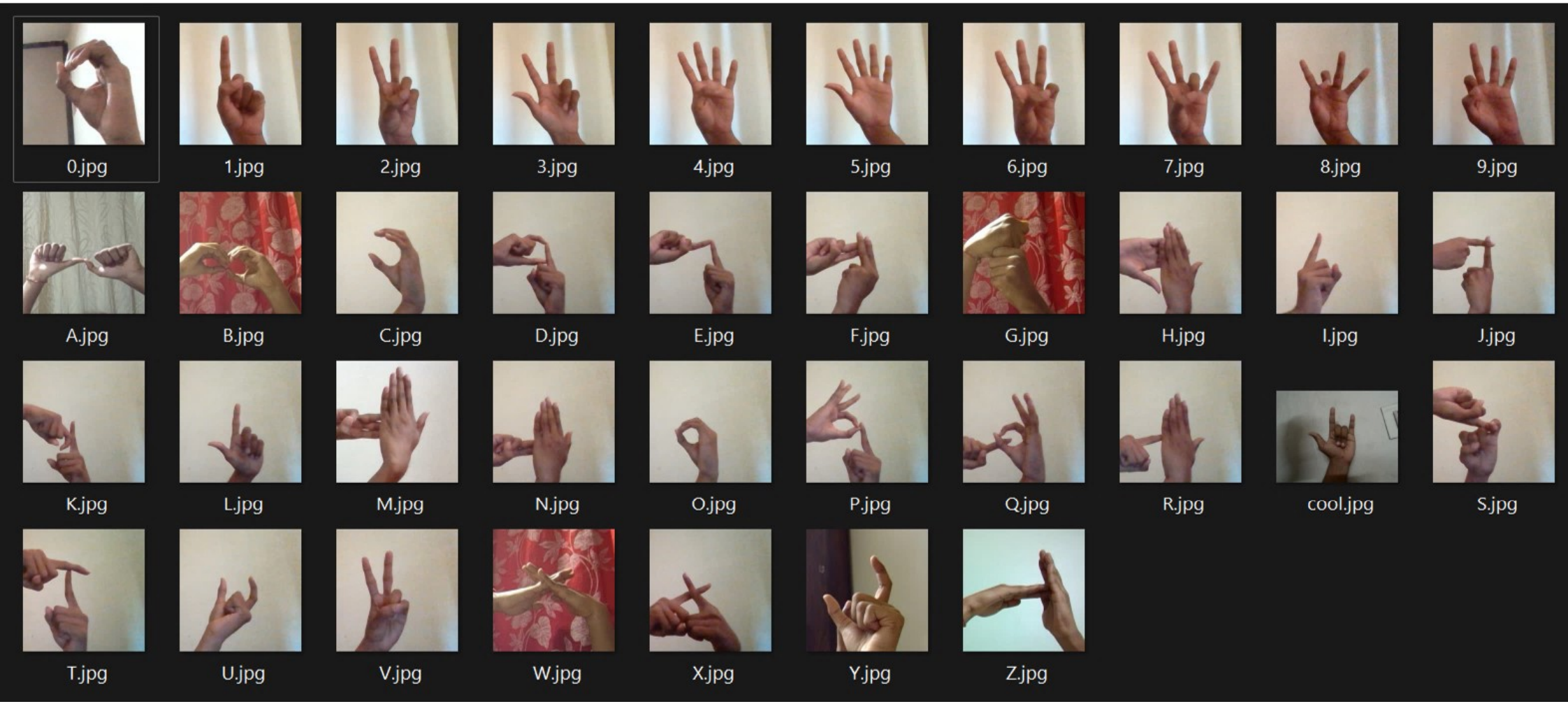
```
score = accuracy_score(y_predict, y_test)

print('{}% of samples were classified correctly !'.format(score * 100))
```

```
98.04618117229128% of samples were classified correctly !
```

```python
acc = accuracy_score(y_predict, y_test)
prec = precision_score(y_predict, y_test, average='macro',zero_division=1)
rec = recall_score(y_predict, y_test, average='macro',zero_division=1)
f1 = f1_score(y_predict, y_test, average='macro')
print("Accuracy:", acc)
print("Precision:", prec)
print("Recall:", rec)
print("F1-score:", f1)
```

```
Accuracy: 0.9804618117229129
Precision: 0.9442624942624942
Recall: 0.9720992769773258
F1-score: 0.9420495585218102
```

```
# Load the data from the pickle file
data_dict = pickle.load(open('./data.pickle', 'rb'))

# Pad sequences to a fixed length
max_length = 100
padded_data = pad_sequences(data_dict['data'], maxlen=max_length,
                            padding='post', dtype='float32', truncating='post')

# Convert data and labels to NumPy arrays
data = np.asarray(padded_data)
labels = np.asarray(data_dict['labels'])

x_train, x_test, y_train, y_test = train_test_split(data, labels, test_size=0.2,
                                                    shuffle=True, stratify=labels, random_state=25)

model = SVC()
# training data
model.fit(x_train, y_train)
# test data
y_predict = model.predict(x_test)


accuracy = accuracy_score(y_test, y_predict)
precision = precision_score(y_test, y_predict, average='macro', zero_division=1)
recall = recall_score(y_test, y_predict, average='macro', zero_division=1)
f1 = f1_score(y_test, y_predict, average='macro')
print('Accuracy: {:.2f}%'.format(accuracy))
print('Precision: {:.2f}'.format(precision))
print('Recall: {:.2f}'.format(recall))
print('F1-score: {:.2f}'.format(f1))

# Save the model
with open('model_svc.p', 'wb') as f:
    pickle.dump({'model': model}, f)
```


0.jpg
1.jpg
2.jpg
3.jpg
4.jpg
5.jpg
6.jpg
7.jpg
8.jpg
9.jpg
A.jpg
B.jpg
C.jpg
D.jpg
E.jpg
F.jpg
G.jpg
H.jpg
I.jpg
J.jpg
K.jpg
L.jpg
M.jpg
N.jpg
O.jpg
P.jpg
Q.jpg
R.jpg
cool.jpg
S.jpg
T.jpg
U.jpg
V.jpg
W.jpg
X.jpg
Y.jpg
Z.jpg

File Edit Selection ··· Untitled (Workspace)

EXPLORER

UNTITLED (WORKSPACE)

- BDA Project
- new
  - .checkpoints
  - data
  - sample
  - collect_imgs.py
  - create_dataset.py
  - data.pickle
  - data1.pickle
  - inference_classifier.py 1
  - model_svc.p
  - model.p
  - model1.p
  - requirements.txt
  - train_classifier.py
  - train_model.ipynb

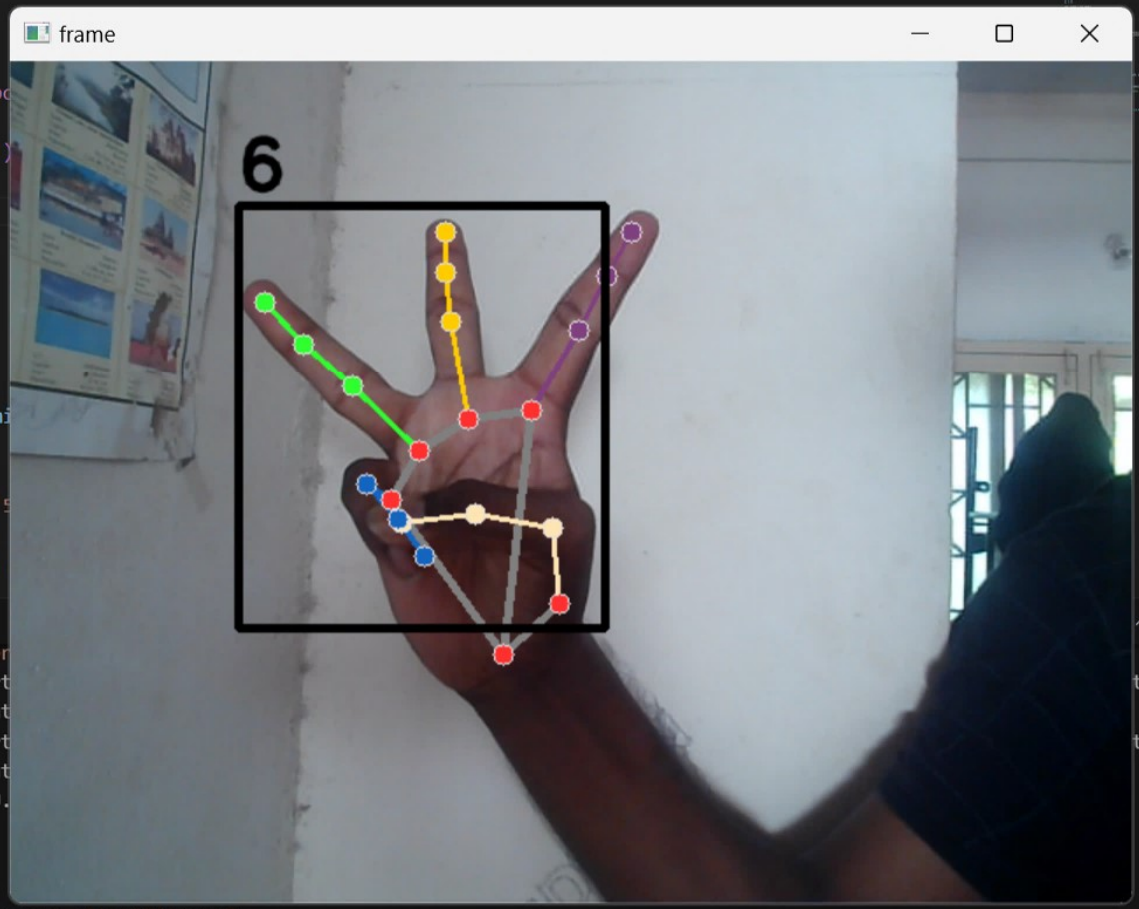
OUTLINE

TIMELINE

collect_imgs.py | create_dataset.py | train_classifier.py | train_model.ipynb | inference_classifier.py 1

new > inference_classifier.py > ...

```python
import pickle
import cv2
import time
import mediapipe as mp
import numpy as np
from tensorflow.keras.preprocessing.sequence impo

model_dict = pickle.load(open('./model1.p', 'rb'))
model = model_dict['model']

cap = cv2.VideoCapture(0)

mp_hands = mp.solutions.hands
mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles

hands = mp_hands.Hands(static_image_mode=True, mi

labels_dict = {
    0: '0', 1: '1', 2: '2', 3: '3', 4: '4', 5: '5
    10: 'A', 11: 'B', 12: 'C', 13: 'D', 14: 'E',
    19: 'J', 20: 'K', 21: 'L', 22: 'M', 23: 'N',
    28: 'S', 29: 'T', 30: 'U', 31: 'V', 32: 'W'
```



PROBLEMS 1 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS

```
[Running] python -u "c:\Users\sanja\Downloads\new\infer
2024-05-06 12:31:12.714660: I tensorflow/core/util/port
floating-point round-off errors from different computat
2024-05-06 12:31:14.213229: I tensorflow/core/util/port
floating-point round-off errors from different computat
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.

[Done] exited with code=1 in 15.57 seconds

[Running] python -u "c:\Users\sanja\Downloads\new\inference_classifier.py"
2024-05-06 12:31:30.057498: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-05-06 12:31:31.613610: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
```

⊗ 0 ⚠ 1 | 0 | Ln 10, Col 1 | Spaces: 4 | UTF-8 | LF | Python | 3.10.9 64-bit | Go Live | Prettier

12:33
06-05-2024

File Edit Selection ··· Untitled (Workspace)

EXPLORER

UNTITLED (WORKSPACE)

- BDA Project
- new
  - .checkpoints
  - data
  - sample
  - collect_imgs.py
  - create_dataset.py
  - data.pickle
  - data1.pickle
  - inference_classifier.py 1
  - model_svc.p
  - model.p
  - model1.p
  - requirements.txt
  - train_classifier.py
  - train_model.ipynb

OUTLINE

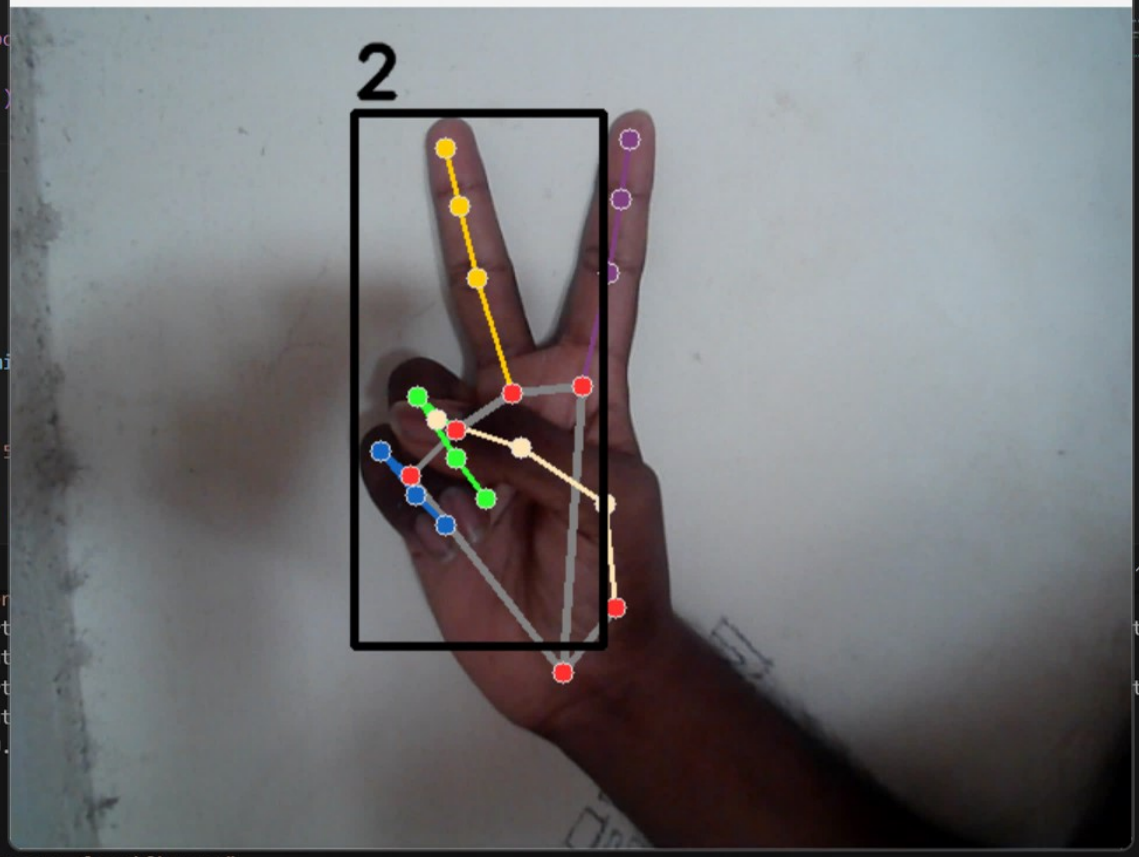
TIMELINE

collect_imgs.py | create_dataset.py | train_classifier.py | train_model.ipynb | inference_classifier.py 1

new > inference_classifier.py > ...

```python
import pickle
import cv2
import time
import mediapipe as mp
import numpy as np
from tensorflow.keras.preprocessing.sequence impo

model_dict = pickle.load(open('./model1.p', 'rb'))
model = model_dict['model']

cap = cv2.VideoCapture(0)

mp_hands = mp.solutions.hands
mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles

hands = mp_hands.Hands(static_image_mode=True, mi

labels_dict = {
    0: '0', 1: '1', 2: '2', 3: '3', 4: '4', 5: '5
    10: 'A', 11: 'B', 12: 'C', 13: 'D', 14: 'E',
    19: 'J', 20: 'K', 21: 'L', 22: 'M', 23: 'N',
    28: 'S', 29: 'T', 30: 'U', 31: 'V', 32: 'W'
```

frame



PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[Running] python -u "c:\Users\sanja\Downloads\new\infer
2024-05-06 12:31:12.714660: I tensorflow/core/util/port
floating-point round-off errors from different computat
2024-05-06 12:31:14.213229: I tensorflow/core/util/port
floating-point round-off errors from different computat
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.

[Done] exited with code=1 in 15.57 seconds

[Running] python -u "c:\Users\sanja\Downloads\new\inference_classifier.py"
2024-05-06 12:31:30.057498: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-05-06 12:31:31.613610: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
```

⊗ 0 ⚠ 1 0 | Ln 10, Col 1 | Spaces: 4 | UTF-8 | LF | Python | 3.10.9 64-bit | Go Live | Prettier

12:33
06-05-2024

File Edit Selection ··· Untitled (Workspace)

EXPLORER

UNTITLED (WORKSPACE)

- BDA Project
- new
  - .checkpoints
  - data
  - sample
  - collect_imgs.py
  - create_dataset.py
  - data.pickle
  - data1.pickle
  - inference_classifier.py 1
  - model_svc.p
  - model.p
  - model1.p
  - requirements.txt
  - train_classifier.py
  - train_model.ipynb

OUTLINE

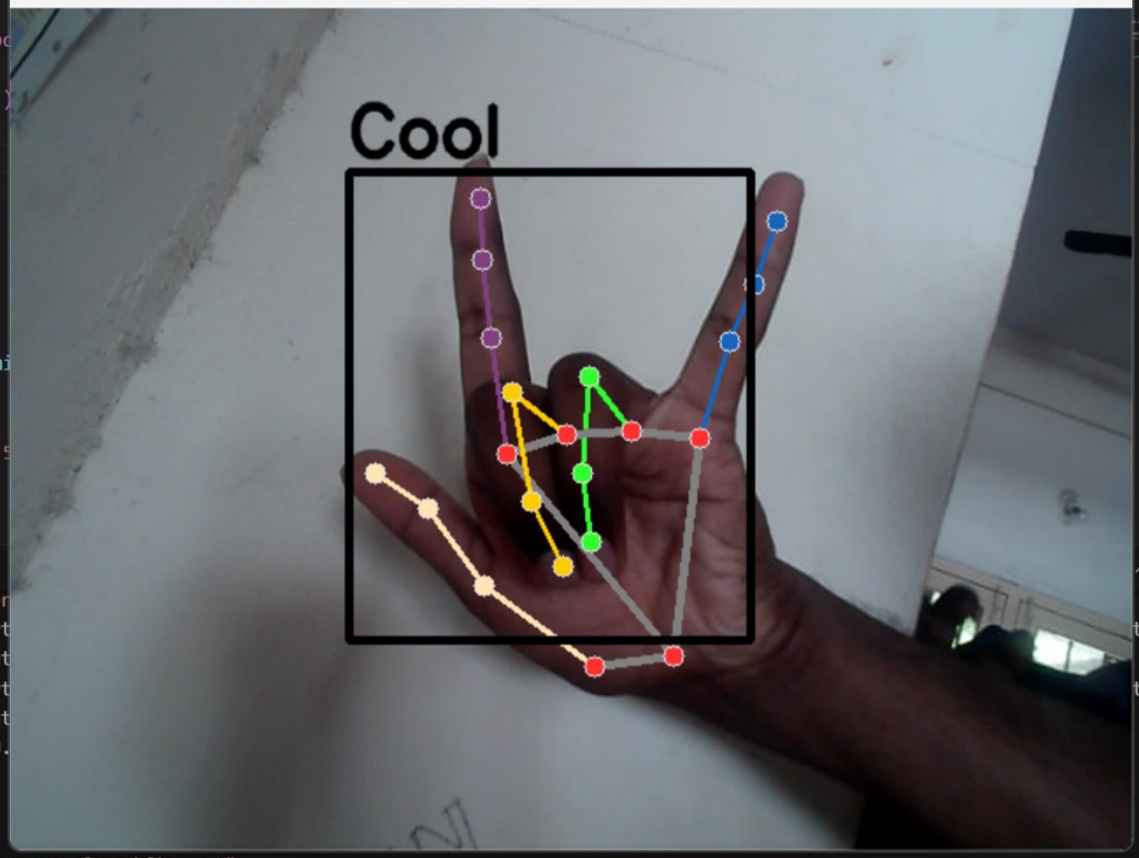
TIMELINE

collect_imgs.py | create_dataset.py | train_classifier.py | train_model.ipynb | inference_classifier.py 1

new > inference_classifier.py > ...

```python
import pickle
import cv2
import time
import mediapipe as mp
import numpy as np
from tensorflow.keras.preprocessing.sequence impo

model_dict = pickle.load(open('./model1.p', 'rb'))
model = model_dict['model']

cap = cv2.VideoCapture(0)

mp_hands = mp.solutions.hands
mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles

hands = mp_hands.Hands(static_image_mode=True, mi

labels_dict = {
    0: '0', 1: '1', 2: '2', 3: '3', 4: '4', 5: '5
    10: 'A', 11: 'B', 12: 'C', 13: 'D', 14: 'E',
    19: 'J', 20: 'K', 21: 'L', 22: 'M', 23: 'N',
    28: 'S', 29: 'T', 30: 'U', 31: 'V', 32: 'W'
```

frame



PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[Running] python -u "c:\Users\sanja\Downloads\new\infer
2024-05-06 12:31:12.714660: I tensorflow/core/util/port
floating-point round-off errors from different computat
2024-05-06 12:31:14.213229: I tensorflow/core/util/port
floating-point round-off errors from different computat
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.

[Done] exited with code=1 in 15.57 seconds

[Running] python -u "c:\Users\sanja\Downloads\new\inference_classifier.py"
2024-05-06 12:31:30.057498: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-05-06 12:31:31.613610: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
```

⊗ 0 ⚠ 1 0 | Ln 10, Col 1 | Spaces: 4 | UTF-8 | LF | Python | 3.10.9 64-bit | Go Live | Prettier

12:34
06-05-2024

File Edit Selection ··· Untitled (Workspace)

EXPLORER

UNTITLED (WORKSPACE)
- BDA Project
- new
  - .checkpoints
  - data
  - sample
  - collect_imgs.py
  - create_dataset.py
  - data.pickle
  - data1.pickle
  - inference_classifier.py 1
  - model_svc.p
  - model.p
  - model1.p
  - requirements.txt
  - train_classifier.py
  - train_model.ipynb

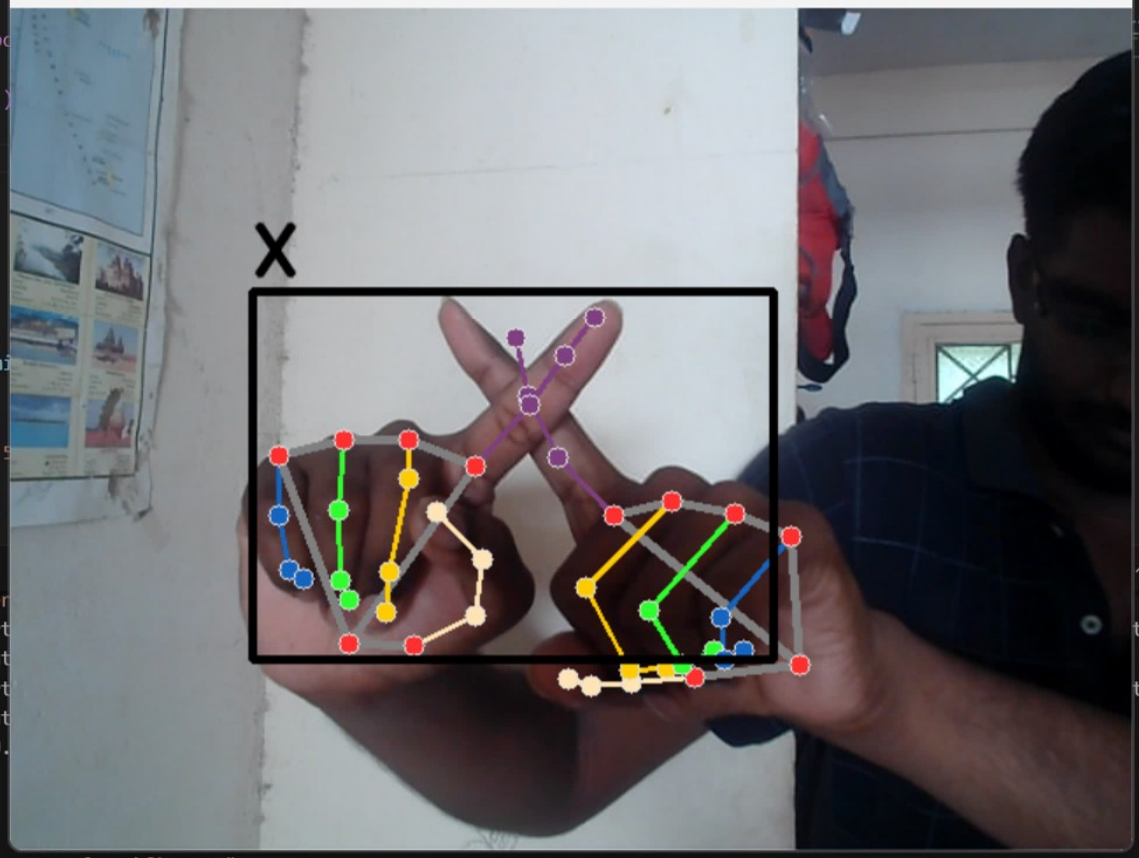
OUTLINE
TIMELINE

collect_imgs.py | create_dataset.py | train_classifier.py | train_model.ipynb | inference_classifier.py 1

new > inference_classifier.py > ... C:\Users\sanja\Downloads\new\create_dataset.py

```python
import pickle
import cv2
import time
import mediapipe as mp
import numpy as np
from tensorflow.keras.preprocessing.sequence impo

model_dict = pickle.load(open('./model1.p', 'rb'))
model = model_dict['model']

cap = cv2.VideoCapture(0)

mp_hands = mp.solutions.hands
mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles

hands = mp_hands.Hands(static_image_mode=True, mi

labels_dict = {
    0: '0', 1: '1', 2: '2', 3: '3', 4: '4', 5: '5
    10: 'A', 11: 'B', 12: 'C', 13: 'D', 14: 'E',
    19: 'J', 20: 'K', 21: 'L', 22: 'M', 23: 'N',
    28: 'S', 29: 'T', 30: 'U', 31: 'V', 32: 'W'
```

frame



PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[Running] python -u "c:\Users\sanja\Downloads\new\infer
2024-05-06 12:31:12.714660: I tensorflow/core/util/port
floating-point round-off errors from different computat
2024-05-06 12:31:14.213229: I tensorflow/core/util/port
floating-point round-off errors from different computat
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.

[Done] exited with code=1 in 15.57 seconds

[Running] python -u "c:\Users\sanja\Downloads\new\inference_classifier.py"
2024-05-06 12:31:30.057498: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-05-06 12:31:31.613610: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
```

⊗ 0 ⚠ 1 0 | Ln 10, Col 1 | Spaces: 4 | UTF-8 | LF | Python | 3.10.9 64-bit | Go Live | Prettier

12:35
06-05-2024

File Edit Selection ··· Untitled (Workspace)

EXPLORER

UNTITLED (WORKSPACE)

- BDA Project
- new
  - .checkpoints
  - data
  - sample
  - collect_imgs.py
  - create_dataset.py
  - data.pickle
  - data1.pickle
  - inference_classifier.py 1
  - model_svc.p
  - model.p
  - model1.p
  - requirements.txt
  - train_classifier.py
  - train_model.ipynb

OUTLINE

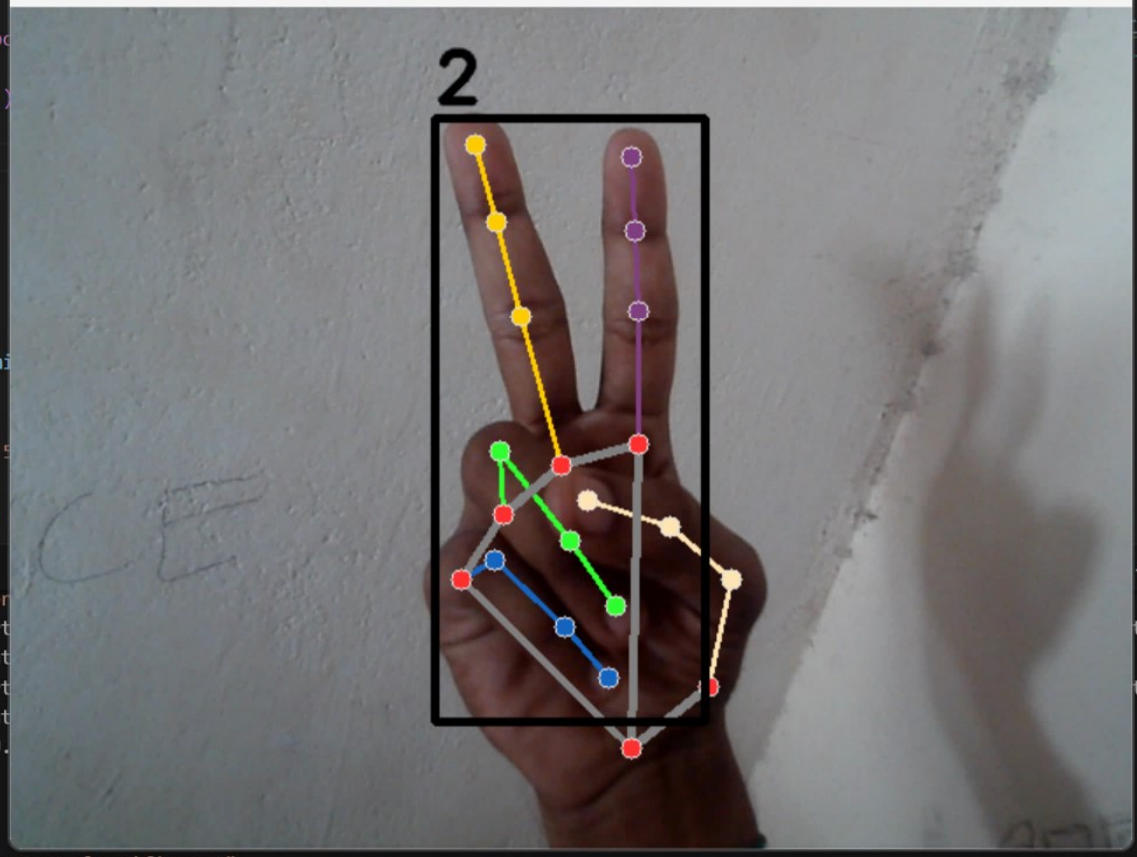
TIMELINE

collect_imgs.py | create_dataset.py | train_classifier.py | train_model.ipynb | inference_classifier.py 1

new > inference_classifier.py > ...

```python
import pickle
import cv2
import time
import mediapipe as mp
import numpy as np
from tensorflow.keras.preprocessing.sequence impo

model_dict = pickle.load(open('./model1.p', 'rb'))
model = model_dict['model']

cap = cv2.VideoCapture(0)

mp_hands = mp.solutions.hands
mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles

hands = mp_hands.Hands(static_image_mode=True, mi

labels_dict = {
    0: '0', 1: '1', 2: '2', 3: '3', 4: '4', 5: '5
    10: 'A', 11: 'B', 12: 'C', 13: 'D', 14: 'E',
    19: 'J', 20: 'K', 21: 'L', 22: 'M', 23: 'N',
    28: 'S', 29: 'T', 30: 'U', 31: 'V', 32: 'W'
```

frame



PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[Running] python -u "c:\Users\sanja\Downloads\new\infer
2024-05-06 12:31:12.714660: I tensorflow/core/util/port
floating-point round-off errors from different computat
2024-05-06 12:31:14.213229: I tensorflow/core/util/port
floating-point round-off errors from different computat
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.

[Done] exited with code=1 in 15.57 seconds

[Running] python -u "c:\Users\sanja\Downloads\new\inference_classifier.py"
2024-05-06 12:31:30.057498: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-05-06 12:31:31.613610: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
```

⊗ 0 ⚠ 1 0 | Ln 10, Col 1 | Spaces: 4 | UTF-8 | LF | Python | 3.10.9 64-bit | Go Live | Prettier

12:37
06-05-2024

File Edit Selection ··· Untitled (Workspace)

EXPLORER

UNTITLED (WORKSPACE)

- BDA Project
- new
  - .checkpoints
  - data
  - sample
  - collect_imgs.py
  - create_dataset.py
  - data.pickle
  - data1.pickle
  - inference_classifier.py 1
  - model_svc.p
  - model.p
  - model1.p
  - requirements.txt
  - train_classifier.py
  - train_model.ipynb

OUTLINE

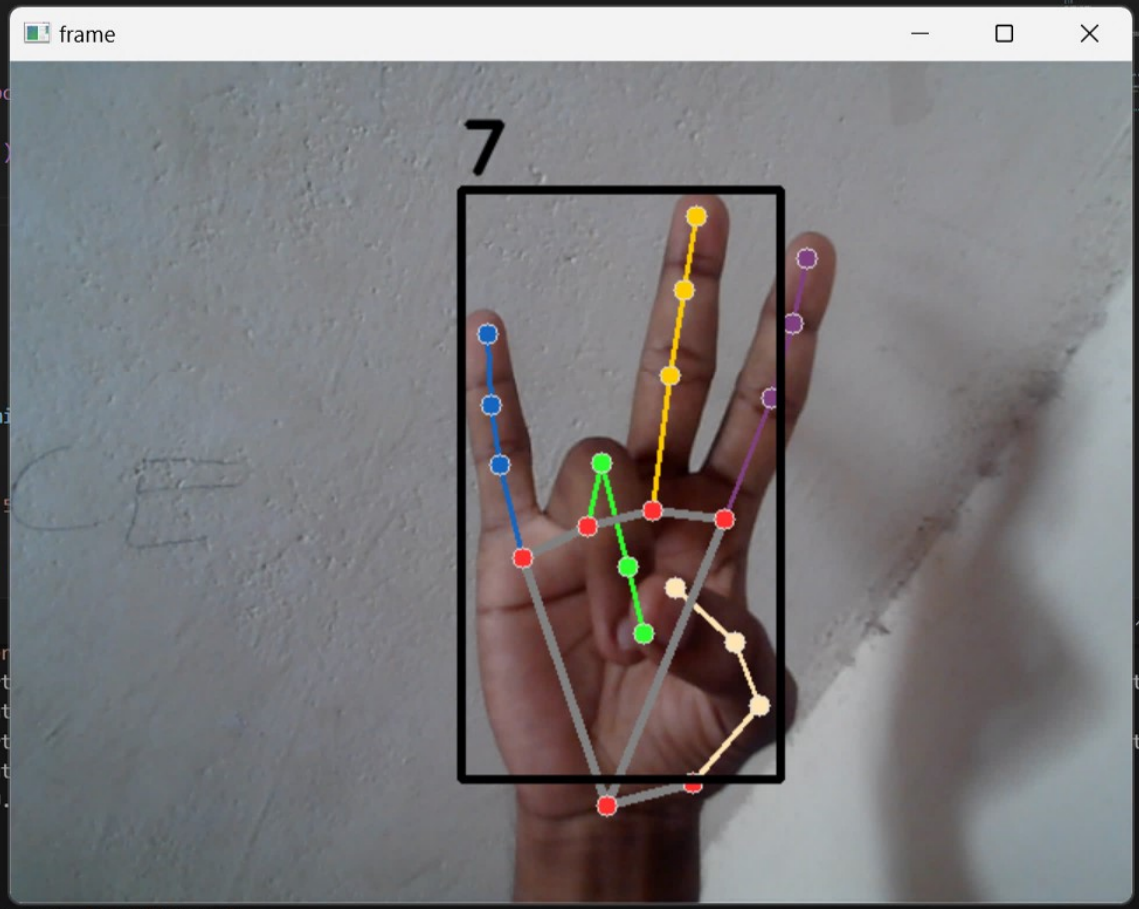
TIMELINE

collect_imgs.py | create_dataset.py | train_classifier.py | train_model.ipynb | inference_classifier.py 1

new > inference_classifier.py > ...

```python
import pickle
import cv2
import time
import mediapipe as mp
import numpy as np
from tensorflow.keras.preprocessing.sequence impo

model_dict = pickle.load(open('./model1.p', 'rb'))
model = model_dict['model']

cap = cv2.VideoCapture(0)

mp_hands = mp.solutions.hands
mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles

hands = mp_hands.Hands(static_image_mode=True, mi

labels_dict = {
    0: '0', 1: '1', 2: '2', 3: '3', 4: '4', 5: '5
    10: 'A', 11: 'B', 12: 'C', 13: 'D', 14: 'E',
    19: 'J', 20: 'K', 21: 'L', 22: 'M', 23: 'N',
    28: 'S', 29: 'T', 30: 'U', 31: 'V', 32: 'W'
```

frame



PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[Running] python -u "c:\Users\sanja\Downloads\new\infer
2024-05-06 12:31:12.714660: I tensorflow/core/util/port
floating-point round-off errors from different computat
2024-05-06 12:31:14.213229: I tensorflow/core/util/port
floating-point round-off errors from different computat
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.

[Done] exited with code=1 in 15.57 seconds

[Running] python -u "c:\Users\sanja\Downloads\new\inference_classifier.py"
2024-05-06 12:31:30.057498: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-05-06 12:31:31.613610: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
```

⊗ 0 ⚠ 1 0 | Ln 10, Col 1 | Spaces: 4 | UTF-8 | LF | Python | 3.10.9 64-bit | Go Live | Prettier

12:37
06-05-2024

File Edit Selection ··· Untitled (Workspace)

EXPLORER

UNTITLED (WORKSPACE)

- BDA Project
- new
  - .checkpoints
  - data
  - sample
  - collect_imgs.py
  - create_dataset.py
  - data.pickle
  - data1.pickle
  - inference_classifier.py 1
  - model_svc.p
  - model.p
  - model1.p
  - requirements.txt
  - train_classifier.py
  - train_model.ipynb

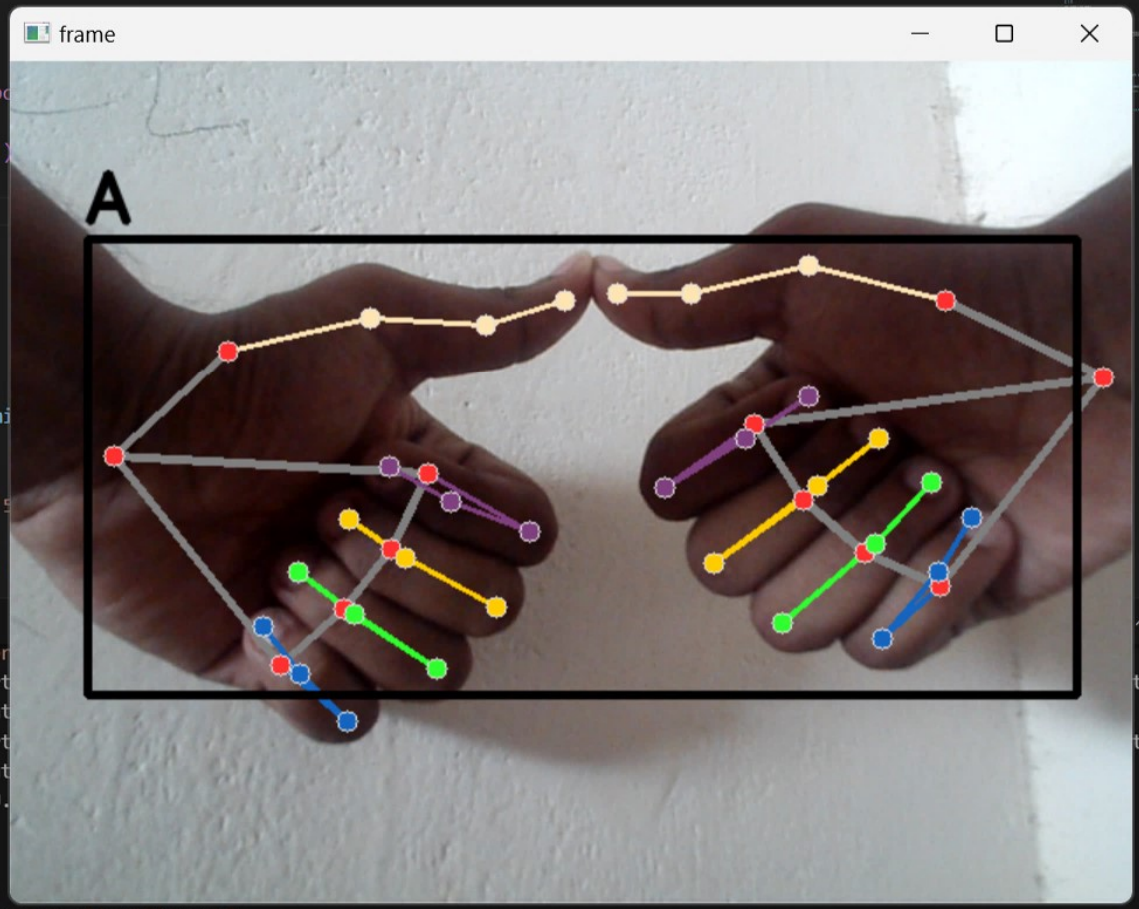
OUTLINE

TIMELINE

collect_imgs.py | create_dataset.py | train_classifier.py | train_model.ipynb | inference_classifier.py 1

new > inference_classifier.py > ...

```python
import pickle
import cv2
import time
import mediapipe as mp
import numpy as np
from tensorflow.keras.preprocessing.sequence impo

model_dict = pickle.load(open('./model1.p', 'rb'))
model = model_dict['model']

cap = cv2.VideoCapture(0)

mp_hands = mp.solutions.hands
mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles

hands = mp_hands.Hands(static_image_mode=True, mi

labels_dict = {
    0: '0', 1: '1', 2: '2', 3: '3', 4: '4', 5: '5
    10: 'A', 11: 'B', 12: 'C', 13: 'D', 14: 'E',
    19: 'J', 20: 'K', 21: 'L', 22: 'M', 23: 'N',
    28: 'S', 29: 'T', 30: 'U', 31: 'V', 32: 'W'
```

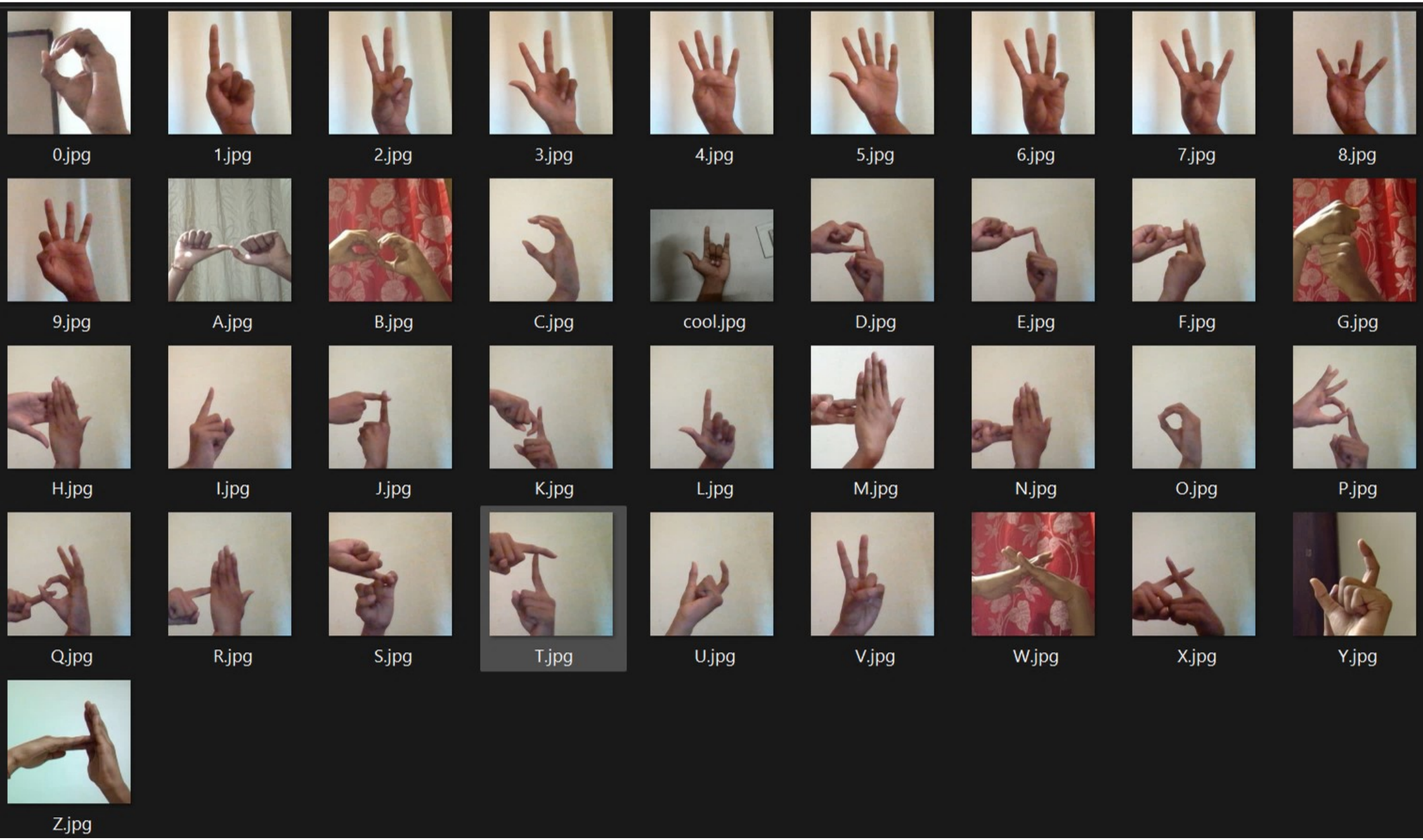


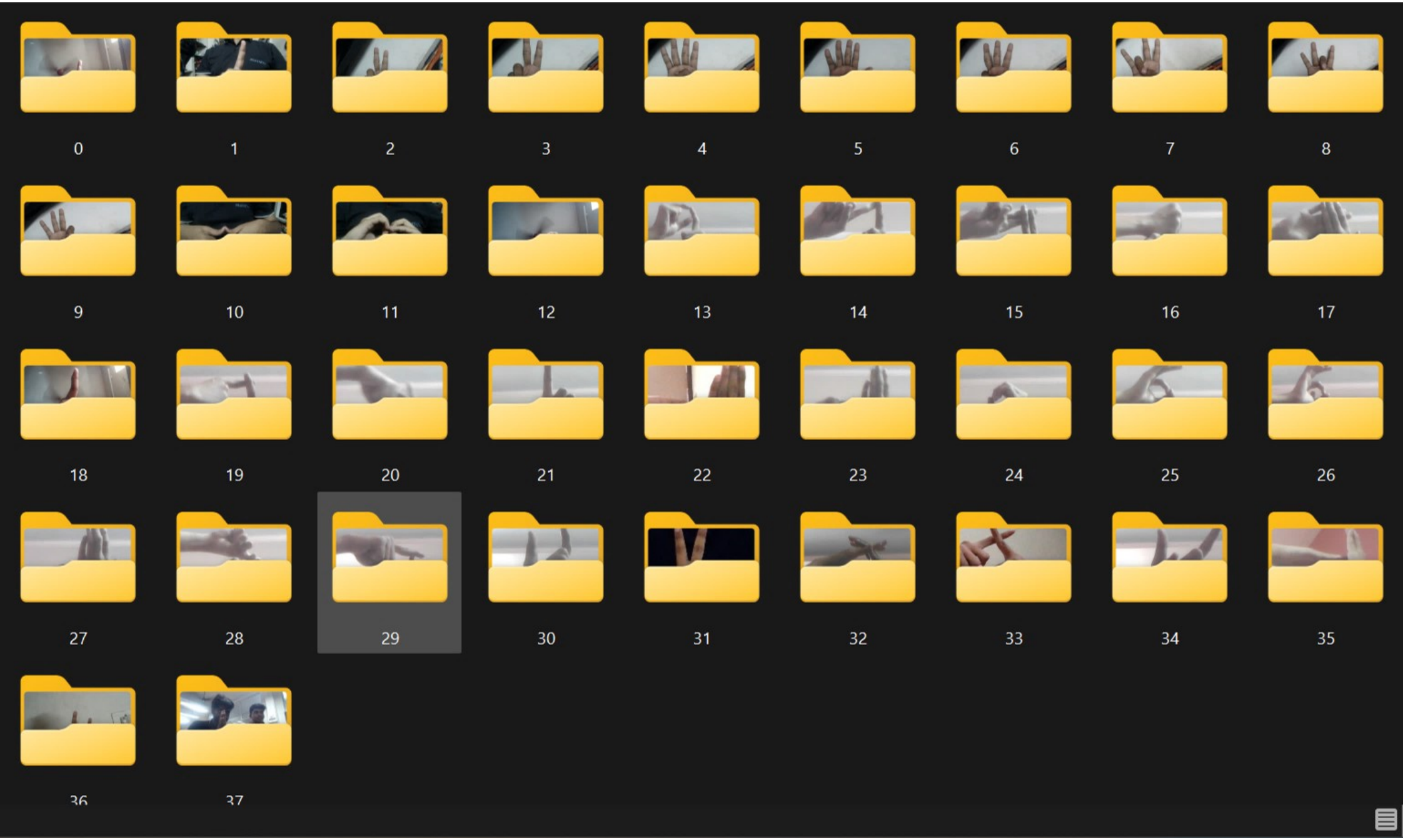
PROBLEMS 1 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS

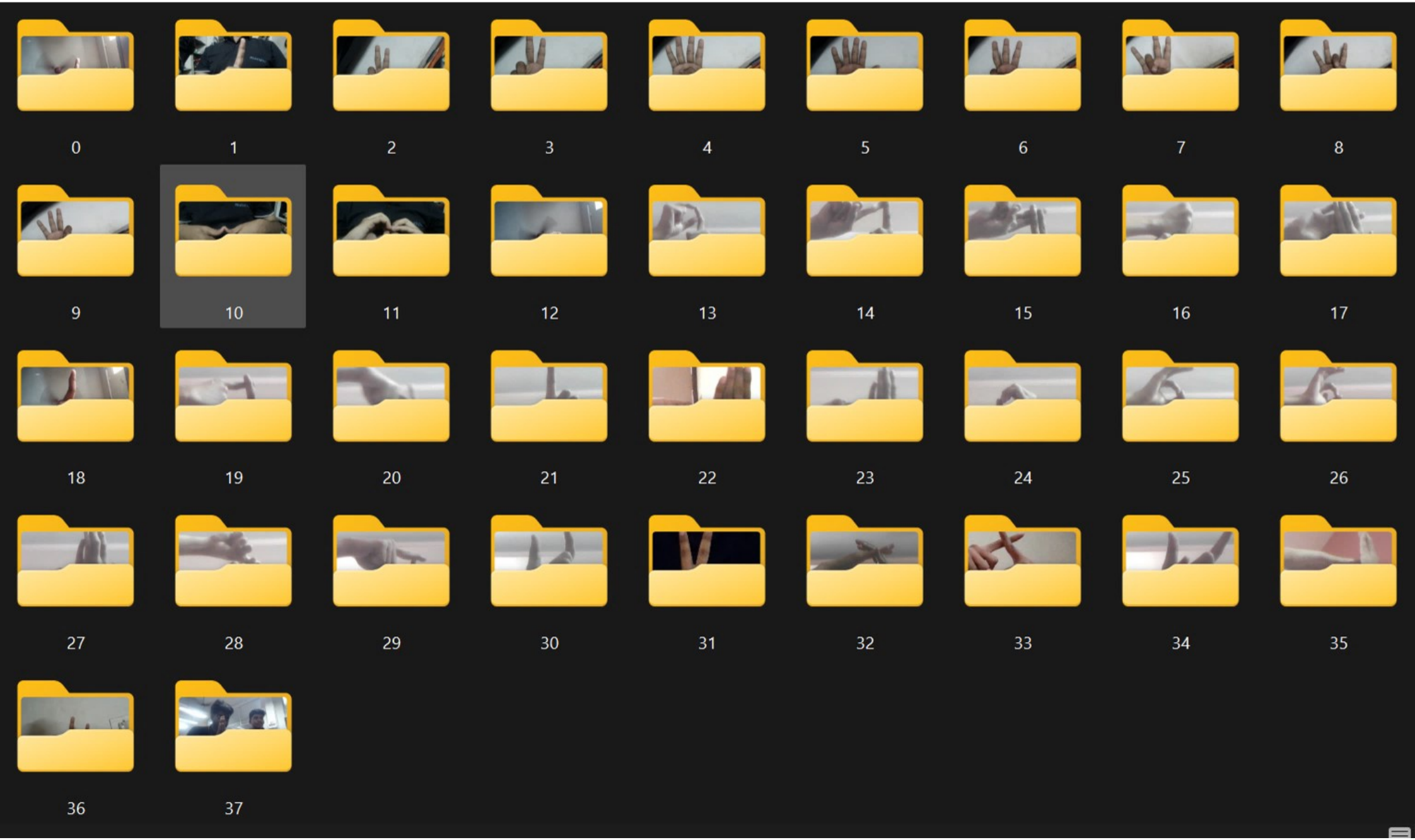
```
[Running] python -u "c:\Users\sanja\Downloads\new\infer
2024-05-06 12:31:12.714660: I tensorflow/core/util/port
floating-point round-off errors from different computat
2024-05-06 12:31:14.213229: I tensorflow/core/util/port
floating-point round-off errors from different computat
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.

[Done] exited with code=1 in 15.57 seconds

[Running] python -u "c:\Users\sanja\Downloads\new\inference_classifier.py"
2024-05-06 12:31:30.057498: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-05-06 12:31:31.613610: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
```

⊗ 0 ⚠ 1 | 0 | Ln 10, Col 1 | Spaces: 4 | UTF-8 | LF | Python | 3.10.9 64-bit | Go Live | Prettier

12:38
06-05-2024


0.jpg
1.jpg
2.jpg
3.jpg
4.jpg
5.jpg
6.jpg
7.jpg
8.jpg
9.jpg
A.jpg
B.jpg
C.jpg
cool.jpg
D.jpg
E.jpg
F.jpg
G.jpg
H.jpg
I.jpg
J.jpg
K.jpg
L.jpg
M.jpg
N.jpg
O.jpg
P.jpg
Q.jpg
R.jpg
S.jpg
T.jpg
U.jpg
V.jpg
W.jpg
X.jpg
Y.jpg
Z.jpg


0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37


0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37

```
x_train, x_test, y_train, y_test = train_test_split(data, labels, test_size=0.2,
                                                    shuffle=True, stratify=labels,random_state=25)

model = RandomForestClassifier()
epochs = 20
for epoch in range(epochs):
    model.fit(x_train, y_train)

y_predict = model.predict(x_test)

score = accuracy_score(y_predict, y_test)
```

new > collect_imgs.py > ...

```python
import os
import cv2
DATA_DIR = './data'
if not os.path.exists(DATA_DIR):
    os.makedirs(DATA_DIR)

number_of_classes = 37
dataset_size = 200
cap = cv2.VideoCapture(0)

for j in range(number_of_classes):
    class_folder_name = str(j)  # Convert integer index to corresponding ASCII character
    class_folder_path = os.path.join(DATA_DIR, class_folder_name)
    if not os.path.exists(class_folder_path):
        os.makedirs(class_folder_path)

    print('Collecting data for class {}'.format(j+37))

    done = False
    while True:
        ret, frame = cap.read()
        cv2.putText(frame, 'Press "Q" to record', (100, 50), cv2.FONT_HERSHEY_SIMPLEX, 1.3, (100, 255, 255), 3,
                    cv2.LINE_AA)
        cv2.imshow('frame', frame)
        if cv2.waitKey(30) == ord('q'):
            break

    counter = 0
    while counter < dataset_size:
        ret, frame = cap.read()
        cv2.imshow('frame', frame)
        cv2.waitKey(30)
        cv2.imwrite(os.path.join(DATA_DIR, str(j), '{}.jpg'.format(counter)), frame)
        counter += 1

cap.release()
cv2.destroyAllWindows()
```

```
import os
import cv2
DATA_DIR = './data'
if not os.path.exists(DATA_DIR):
    os.makedirs(DATA_DIR)

number_of_classes = 37
dataset_size = 200
cap = cv2.VideoCapture(0)

for j in range(number_of_classes):
    class_folder_name = str(j)  # Convert integer index to corresponding ASCII character
    class_folder_path = os.path.join(DATA_DIR, class_folder_name)
    if not os.path.exists(class_folder_path):
        os.makedirs(class_folder_path)

    print('Collecting data for class {}'.format(j+37))

    done = False
    while True:
        ret, frame = cap.read()
        cv2.putText(frame, 'Press "Q" to record', (100, 50), cv2.FONT_HERSHEY_SIMPLEX, 1.3,
                    (100, 255, 255), 3, cv2.LINE_AA)
        cv2.imshow('frame', frame)
        if cv2.waitKey(30) == ord('q'):
            break

    counter = 0
    while counter < dataset_size:
        ret, frame = cap.read()
        cv2.imshow('frame', frame)
        cv2.waitKey(30)
        cv2.imwrite(os.path.join(DATA_DIR, str(j), '{}.jpg'.format(counter)), frame)
        counter += 1

cap.release()
cv2.destroyAllWindows()
```

```
data = []
labels = []
for dir_ in os.listdir(DATA_DIR):
    for img_path in os.listdir(os.path.join(DATA_DIR, dir_)):
        data_aux = []

        x_ = []
        y_ = []

        img = cv2.imread(os.path.join(DATA_DIR, dir_, img_path))
        img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

        results = hands.process(img_rgb)
        if results.multi_hand_landmarks:
            for hand_landmarks in results.multi_hand_landmarks:
                for i in range(len(hand_landmarks.landmark)):
                    x = hand_landmarks.landmark[i].x
                    y = hand_landmarks.landmark[i].y

                    x_.append(x)
                    y_.append(y)

                for i in range(len(hand_landmarks.landmark)):
                    x = hand_landmarks.landmark[i].x
                    y = hand_landmarks.landmark[i].y
                    data_aux.append(x - min(x_))
                    data_aux.append(y - min(y_))

            data.append(data_aux)
            labels.append(dir_)

f = open('data1.pickle', 'wb')
pickle.dump({'data': data, 'labels': labels}, f)
f.close()
```

```
data_dict = pickle.load(open('./data1.pickle', 'rb'))

# Pad sequences to a fixed length
max_length = 100  # Set your desired maximum length
padded_data = pad_sequences(data_dict['data'], maxlen=max_length, padding='post',
                            dtype='float32', truncating='post')

data = np.asarray(padded_data)
labels = np.asarray(data_dict['labels'])

x_train, x_test, y_train, y_test = train_test_split(data, labels, test_size=0.2,
                                                    shuffle=True, stratify=labels,random_state=25)

model = RandomForestClassifier()
epochs = 20
for epoch in range(epochs):
    model.fit(x_train, y_train)

y_predict = model.predict(x_test)

score = accuracy_score(y_predict, y_test)

print('{}% of samples were classified correctly !'.format(score * 100))

acc = accuracy_score(y_predict, y_test)
prec = precision_score(y_predict, y_test, average='macro',zero_division=1)
rec = recall_score(y_predict, y_test, average='macro',zero_division=1)
f1 = f1_score(y_predict, y_test, average='macro')
print("Accuracy:", acc)
print("Precision:", prec)
print("Recall:", rec)
print("F1-score:", f1)

f = open('model1.p', 'wb')
pickle.dump({'model': model}, f)
f.close()
```