

```
In [1]: import tensorflow as tf from
tensorflow import keras import
numpy as np import
matplotlib.pyplot as plt
```

```
In [2]: fashion_mnist=keras.datasets.fashion_mnist
(train_images,train_labels),(test_images,test_labels)=fashion_mnist.load_data()
train_images=train_images/255.0 test_images=test_images/255.0
```

```
In [3]: model=keras.Sequential([    keras.layers.Conv2D(64,(3,3),activation='relu'    ,
    input_shape=(28,28,1)),          keras.layers.MaxPooling2D((2,2)),
    keras.layers.Flatten(),          keras.layers.Dense(128,    activation='relu'),
    keras.layers.Dense(10,activation='softmax')
])
```

```
In [4]: model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
    metrics=['accuracy'])
```

```
In [5]: model.fit(train_images,train_labels,epochs=10)
```

Epoch 1/10
 1875/1875 ————— 70s 36ms/step - accuracy: 0.8214 - loss: 0.4950

Epoch 2/10
 1875/1875 ————— 63s 34ms/step - accuracy: 0.9068 - loss: 0.2560

Epoch 3/10
 1875/1875 ————— 82s 34ms/step - accuracy: 0.9277 - loss: 0.2012

Epoch 4/10
 1875/1875 ————— 83s 34ms/step - accuracy: 0.9399 - loss: 0.1613

Epoch 5/10
 1875/1875 ————— 64s 34ms/step - accuracy: 0.9496 - loss: 0.1339

Epoch 6/10
 1875/1875 ————— 61s 33ms/step - accuracy: 0.9583 - loss: 0.1122

Epoch 7/10
 1875/1875 ————— 188s 100ms/step - accuracy: 0.9656 - loss: 0.0939

Epoch 8/10
 1875/1875 ————— 63s 33ms/step - accuracy: 0.9716 - loss: 0.0760

Epoch 9/10
 1875/1875 ————— 82s 33ms/step - accuracy: 0.9784 - loss: 0.0594

Epoch 10/10
 1875/1875 ————— 61s 33ms/step - accuracy: 0.9821 - loss: 0.0494

Out[5]: <keras.src.callbacks.history.History at 0x23c51752150>

In [6]:

```
test_loss,test_acc=model.evaluate(test_images,test_labels) print("test
Accuracy:",test_acc)
```

313/313 ————— 3s 10ms/step - accuracy: 0.9127 - loss: 0.3514 test
 Accuracy: 0.9140999913215637

In [7]:

```
predictions=model.predict(test_images) predicted_labels=np.argmax(predictions,axis=1)
```

313/313 ————— 3s 9ms/step

In [8]:

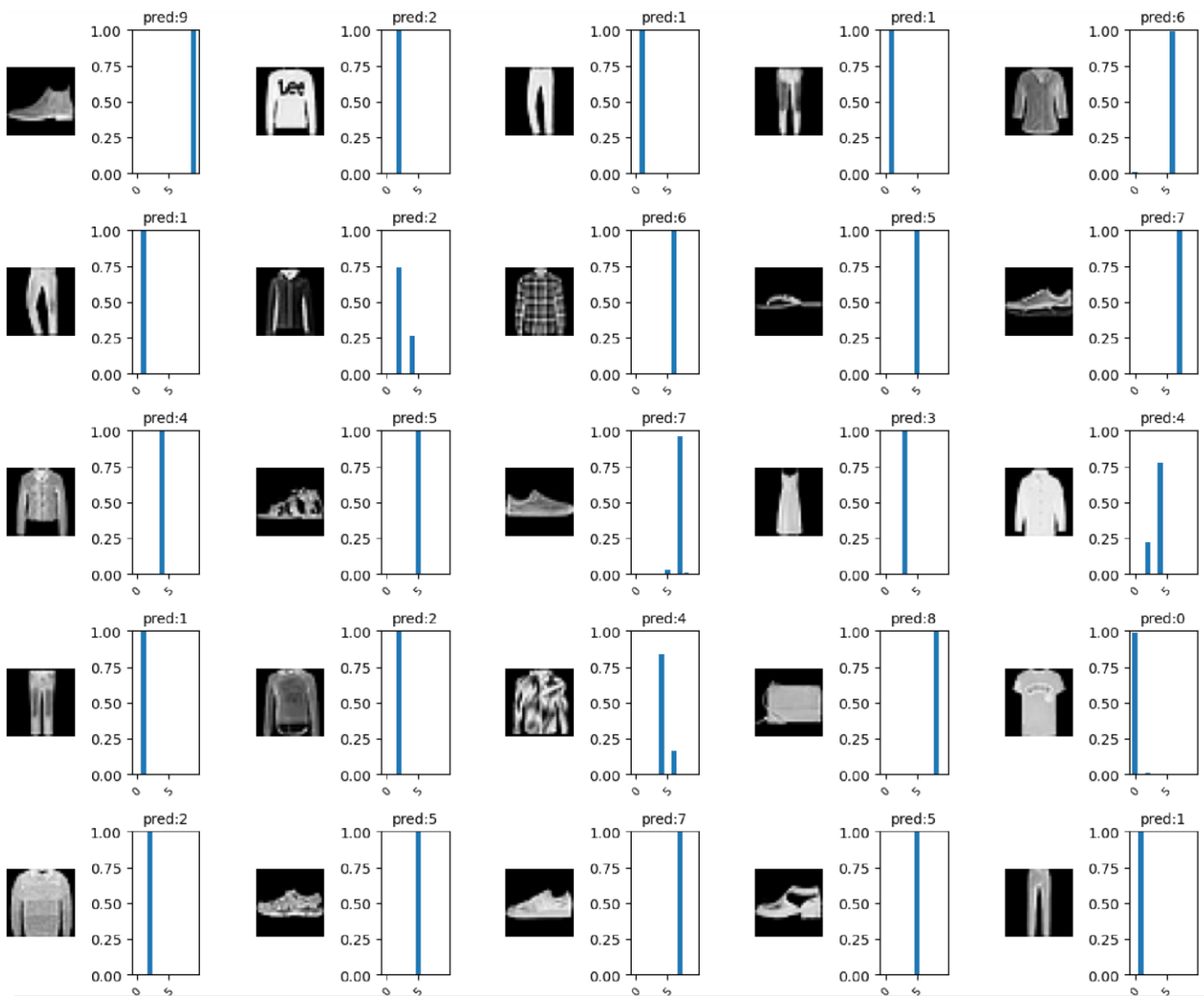
```
import numpy as np import
matplotlib.pyplot as plt
num_rows=5
num_cols=5
num_images=num_rows * num_cols plt.figure(figsize=(12,10))

for i in range(num_images):
    plt.subplot(num_rows,num_cols*2,2*i+1)
    plt.imshow(test_images[i],cmap='grey') plt.axis('off')
    plt.subplot(num_rows,num_cols*2,2*i+2)
```

```
plt.bar(range(10), predictions[i])
```

```
plt.xticks(np.arange(0,10,5), fontsize=8, rotation=45)
plt.yticks(np.arange(0,1.1,0.25))
plt.ylim([0,1])
plt.title(f"pred:{predicted_labels[i]}", fontsize=10)
```

```
plt.tight_layout()
plt.show()
```



```
In [
]:
```