

✓ V6 [BEST ONE SO FAR]

```
!git clone https://github.com/shafidaaaa/Bangkit.git
```

```
fatal: destination path 'Bangkit' already exists and is not an empty directory.
```

```
# Define the path to the images directory
```

```
image_directory1 = '/content/Bangkit/Capstone/bisindo_data/Citra BISINDO'
```

```
image_directory2 = '/content/Bangkit/Capstone/bisindo_data/bisindo_dataset_v2'
```

```
image_directory3 = '/content/Bangkit/Capstone/bisindo_data/bisindo_dataset_v3'
```

VGG16 SOLVED ISSUES

```
!pip uninstall -y tensorflow tensorflowjs tensorflow_decision_forests
```

```
Found existing installation: tensorflow 2.16.1
Uninstalling tensorflow-2.16.1:
  Successfully uninstalled tensorflow-2.16.1
Found existing installation: tensorflowjs 4.20.0
Uninstalling tensorflowjs-4.20.0:
  Successfully uninstalled tensorflowjs-4.20.0
Found existing installation: tensorflow_decision_forests 1.9.1
Uninstalling tensorflow_decision_forests-1.9.1:
  Successfully uninstalled tensorflow_decision_forests-1.9.1
```

```
!pip install tensorflow==2.16.1 tensorflowjs
```

```
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.16.1)
Requirement already satisfied: ml-dtypes~=0.3.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.16.1)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.16.1)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.16.1)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.16.1)
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.16.1)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.16.1)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.16.1)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.16.1)
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.16.1)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.16.1)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.16.1)
```

Requirement already satisfied: tf-keras>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: tensorflow-decision-forests>=1.5.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: tensorflow-hub>=0.16.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: msgpack in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: optax in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: orbax-checkpoint in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: tensorstore in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: rich>=11.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: PyYAML>=5.4.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: scipy>=1.9 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: namex in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: optree in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: wurlitzer in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: ydf in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: chex>=0.1.86 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: etils[epath,epy] in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: nest_asyncio in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: toolz>=0.9.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

Requirement already satisfied: zipp in /usr/local/lib/python3.10/dist-packages (from tensorflow>=2.13.0)

```
import os
import numpy as np
import tensorflow as tf
from tensorflow.keras.applications import VGG16
from tensorflow.keras.preprocessing.image import load_img, img_to_array
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D, Flatten
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint

BATCH_SIZE = 4
IMAGE_SIZE = (224, 224)

# Load images and labels
def load_images_and_labels(image_directory, image_size=IMAGE_SIZE, batch_size=BATCH_SIZE):
    images = []
    labels = []
    for label in os.listdir(image_directory):
        label_path = os.path.join(image_directory, label)
        if os.path.isdir(label_path):
            image_paths = [os.path.join(label_path, image_name) for image_name in os.listdir(label_path)]
            for batch_start in range(0, len(image_paths), batch_size):
                batch_end = min(batch_start + batch_size, len(image_paths))
                batch_image_paths = image_paths[batch_start:batch_end]
                batch_images = []
                for image_path in batch_image_paths:
                    image = tf.keras.preprocessing.image.load_img(image_path, target_size=image_size)
                    image = tf.keras.preprocessing.image.img_to_array(image)
                    image = image / 255.0 # Normalize the image
                    batch_images.append(image)
                images.extend(batch_images)
                labels.extend([label] * len(batch_images))
    return np.array(images), np.array(labels)

# Load images and labels
X1, y1 = load_images_and_labels(image_directory1)
X2, y2 = load_images_and_labels(image_directory2)
X3, y3 = load_images_and_labels(image_directory3)

# Combine the data
X = np.concatenate((X1, X2, X3), axis=0)
y = np.concatenate((y1, y2, y3), axis=0)

# Encode labels
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)
y_categorical = tf.keras.utils.to_categorical(y_encoded)

# Split into training and test sets
```

```

X_train, X_test, y_train, y_test = train_test_split(X, y_categorical, test_size=0.2, random_

# Load VGG16 model pre-trained on ImageNet
base_model = VGG16(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

# Freeze the layers of the base model
for layer in base_model.layers:
    layer.trainable = False

# Create a new model that includes both the VGG16 base model and the classifier
model = Sequential([
    base_model,
    Flatten(),
    Dense(512, activation='relu'),
    Dense(len(label_encoder.classes_), activation='softmax')
])

# Compile the model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Define callbacks
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
model_checkpoint = ModelCheckpoint('best_model.keras', save_best_only=True)

# Train the model
history = model.fit(X_train, y_train,
                    batch_size=BATCH_SIZE,
                    epochs=10,
                    validation_data=(X_test, y_test),
                    callbacks=[early_stopping, model_checkpoint]
)

```



Epoch 1/10

321/321 ————— 19s 54ms/step - accuracy: 0.3214 - loss: 3.9289 - val_accu

Epoch 2/10

321/321 ————— 13s 42ms/step - accuracy: 0.9417 - loss: 0.2600 - val_accu

Epoch 3/10

321/321 ————— 20s 42ms/step - accuracy: 0.9820 - loss: 0.0990 - val_accu

Epoch 4/10

321/321 ————— 21s 42ms/step - accuracy: 0.9781 - loss: 0.0761 - val_accu

Epoch 5/10

321/321 ————— 14s 42ms/step - accuracy: 0.9973 - loss: 0.0155 - val_accu

Epoch 6/10

321/321 ————— 14s 44ms/step - accuracy: 1.0000 - loss: 0.0062 - val_accu

Epoch 7/10

321/321 ————— 12s 37ms/step - accuracy: 1.0000 - loss: 0.0027 - val_accu

Epoch 8/10

321/321 ————— 20s 37ms/step - accuracy: 1.0000 - loss: 0.0022 - val_accu

Epoch 9/10

321/321 ————— 21s 39ms/step - accuracy: 1.0000 - loss: 0.0017 - val_accu

Epoch 10/10

321/321 ————— 14s 43ms/step - accuracy: 1.0000 - loss: 0.0015 - val_accu

```
# Save model
model.save('modelVGG.h5')

# Evaluate model
test_loss, test_acc = model.evaluate(X_test, y_test)
print(f"Test accuracy: {test_acc:.4f}")

# Make predictions
predictions = model.predict(X_test)

# Predicted classes
predicted_classes = np.argmax(predictions, axis=1)
predicted_class = label_encoder.inverse_transform(predicted_classes)

# True classes
true_classes = np.argmax(y_test, axis=1)
true_class = label_encoder.inverse_transform(true_classes)

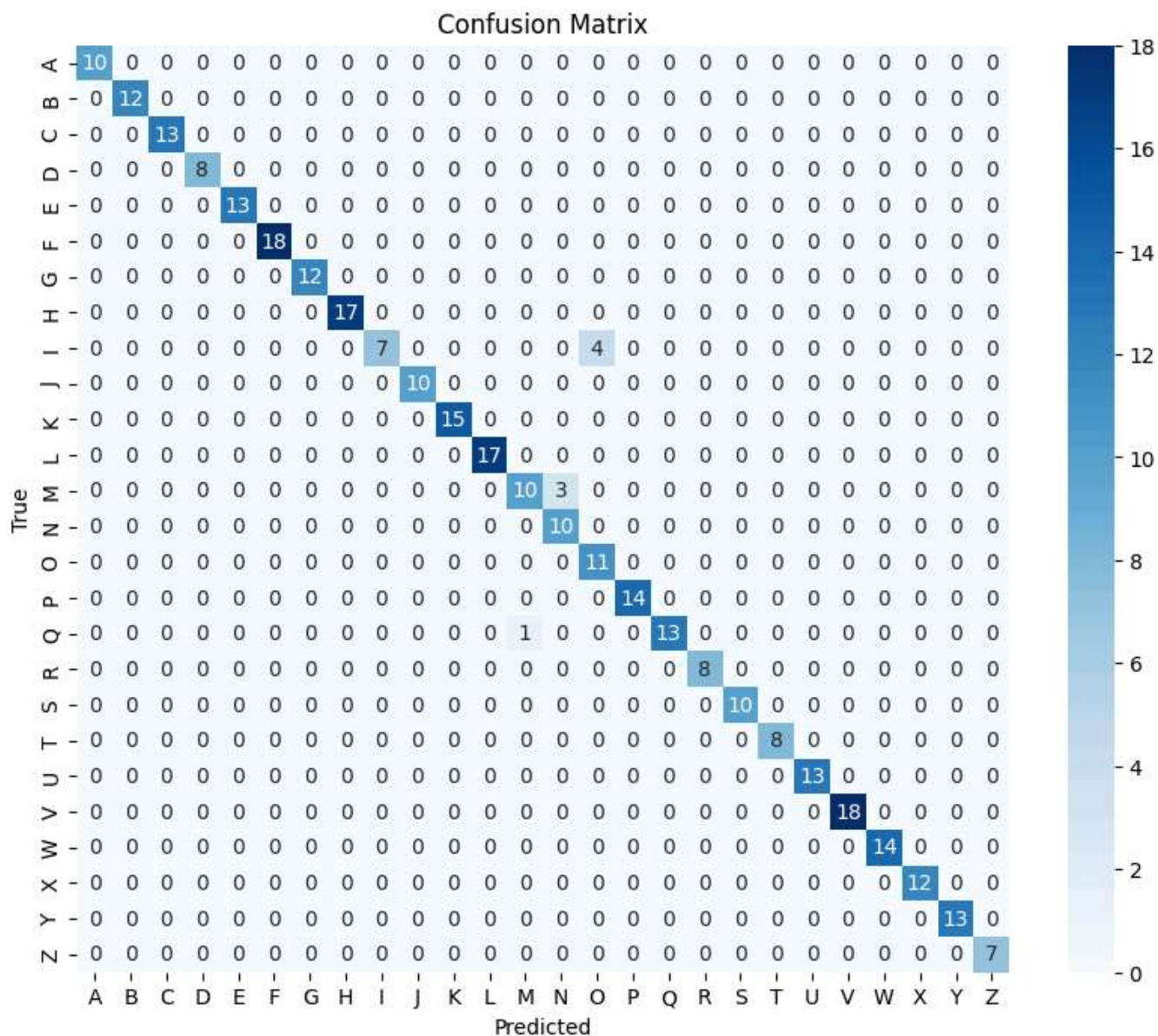
print("Predicted classes:", predicted_class[:10])
print("True classes      :", true_class[:10])
```

⚠ WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.savi`

11/11 ————— 2s 121ms/step - accuracy: 0.9694 - loss: 0.1138
 Test accuracy: 0.9751
 11/11 ————— 2s 154ms/step
 Predicted classes: ['V' 'R' 'V' 'Z' 'U' 'G' 'B' 'S' 'Z' 'O']
 True classes : ['V' 'R' 'V' 'Z' 'U' 'G' 'B' 'S' 'Z' 'O']

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix

# Confusion Matrix
conf_matrix = confusion_matrix(true_classes, predicted_classes)
plt.figure(figsize=(10, 8))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=label_encoder.classe
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()
```



```
plt.figure(figsize=(14, 5))
```

```
# Accuracy plot
```

```
plt.subplot(1, 2, 1)
```

```
plt.plot(history.history['accuracy'], label='Train Accuracy')
```

```
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
```

```
plt.xlabel('Epochs')
```

```
plt.ylabel('Accuracy')
```

```
plt.legend()
```

```
plt.title('Accuracy over Epochs')
```

```
# Loss plot
```

```
plt.subplot(1, 2, 2)
```

```
plt.plot(history.history['loss'], label='Train Loss')
```

```
plt.plot(history.history['val_loss'], label='Validation Loss')
```

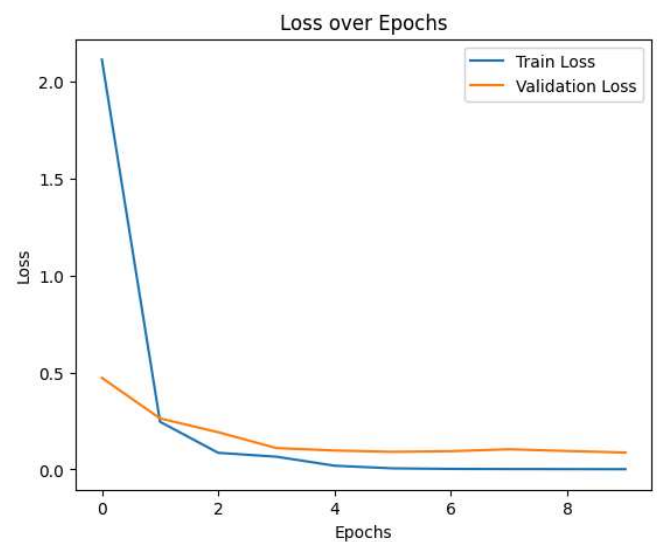
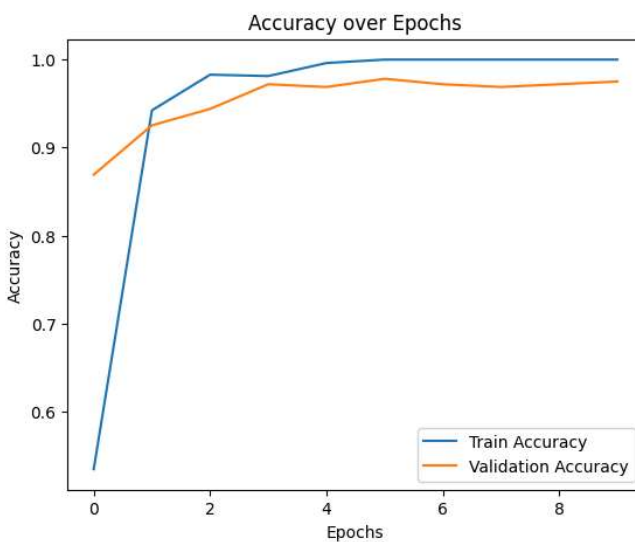
```
plt.xlabel('Epochs')
```

```
plt.ylabel('Loss')
```

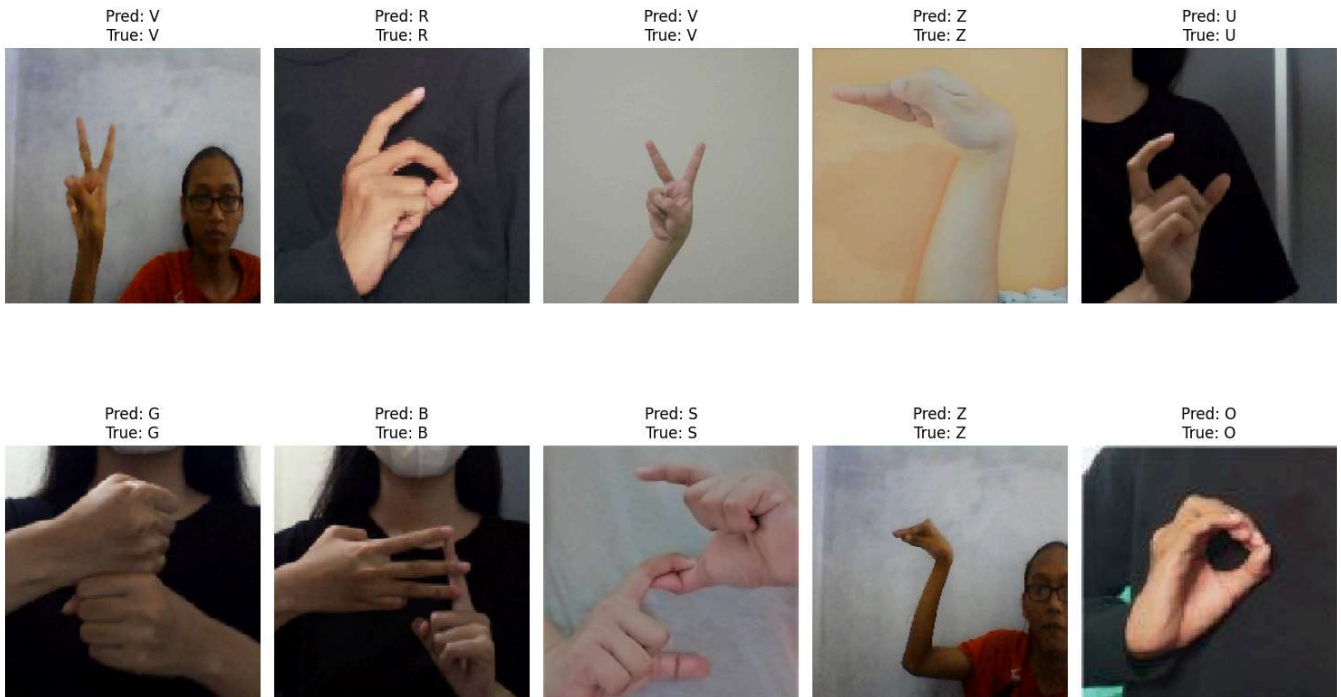
```
plt.legend()
```

```
plt.title('Loss over Epochs')
```

```
plt.show()
```



```
# Visualize a few sample predictions along with their true labels
num_samples = 10
plt.figure(figsize=(15, 10))
for i in range(num_samples):
    plt.subplot(2, 5, i+1)
    plt.imshow(X_test[i])
    plt.title(f"Pred: {predicted_class[i]}\nTrue: {true_class[i]}")
    plt.axis('off')
plt.tight_layout()
plt.show()
```



```
directory = "tfjs_model1"
parent_dir = "/content"
path = os.path.join(parent_dir, directory)
os.makedirs(path, exist_ok=True)
```



```
import tensorflowjs as tfjs
```

```
!tensorflowjs_converter --input_format keras modelVGG.h5 /content/tfjs_model
```

2024-06-09 18:08:19.435670: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-1 failed to lookup keras version from the file, this is likely a weight only file

```
!ls /content/tfjs_model
```

```
group1-shard10of27.bin  group1-shard17of27.bin  group1-shard23of27.bin  group1-shard4of27.bin
group1-shard11of27.bin  group1-shard18of27.bin  group1-shard24of27.bin  group1-shard5of27.bin
group1-shard12of27.bin  group1-shard19of27.bin  group1-shard25of27.bin  group1-shard6of27.bin
group1-shard13of27.bin  group1-shard1of27.bin   group1-shard26of27.bin  group1-shard7of27.bin
group1-shard14of27.bin  group1-shard20of27.bin  group1-shard27of27.bin  group1-shard8of27.bin
group1-shard15of27.bin  group1-shard21of27.bin  group1-shard2of27.bin   group1-shard9of27.bin
group1-shard16of27.bin  group1-shard22of27.bin  group1-shard3of27.bin   model.json
```

```
!zip -r tfjs_model.zip tfjs_model
from google.colab import files
files.download('tfjs_model.zip')
```

```

➦ adding: tfjs_model/ (stored 0%)
  adding: tfjs_model/group1-shard5of27.bin (deflated 7%)

```

```
import tensorflowjs as tfjs
```

```
!tensorflowjs_converter --input_format keras modelVGG.h5 /content/tfjs_model1
```

```

➦ 2024-06-09 18:18:42.662261: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-1
failed to lookup keras version from the file,
  this is likely a weight only file

```



```

!zip -r tfjs_model1.zip tfjs_model1
from google.colab import files
files.download('tfjs_model1.zip')

```

```

➦ adding: tfjs_model1/ (stored 0%)
  adding: tfjs_model1/group1-shard5of27.bin (deflated 7%)
  adding: tfjs_model1/group1-shard11of27.bin (deflated 7%)
  adding: tfjs_model1/group1-shard19of27.bin (deflated 7%)
  adding: tfjs_model1/group1-shard1of27.bin (deflated 8%)
  adding: tfjs_model1/group1-shard23of27.bin (deflated 7%)
  adding: tfjs_model1/group1-shard10of27.bin (deflated 7%)
  adding: tfjs_model1/group1-shard14of27.bin (deflated 7%)
  adding: tfjs_model1/group1-shard8of27.bin (deflated 7%)
  adding: tfjs_model1/group1-shard25of27.bin (deflated 7%)
  adding: tfjs_model1/group1-shard16of27.bin (deflated 7%)
  adding: tfjs_model1/group1-shard22of27.bin (deflated 7%)
  adding: tfjs_model1/group1-shard4of27.bin (deflated 7%)

```