PERFORMANCE AND FINAL SUBMISSION PHRASE

Date	10 November 2023
Team ID	Team-591587
Project Name	ASL- Alphabet Image Recognition
Maximum Marks	10 Marks

S.No.	Parameter	Values	Screenshot
1	**Metrics**	**Classification Model: VGG16 model Confusion Matrix , Classification Report & Accuracy ScoresTraining accuracy:- 94.98% Testing accuracy:- 96.26%**	**Confusion Matrix:**

```
136/136 [==============================] - 25s 177ms/step
tf.Tensor(
[[580   1   1   0   5   0   0   0   0   0   0   0   5   0   0   0   0   0
    8   0   0   0   0   0   0   0   0   0   0]
 [  1 590   0   0   4   1   0   0   1   0   0   0   0   0   1   0   0   0
    0   0   0   0   0   0   0   0   1   1   0]
 [  0   0 583   0   1   0   0   0   0   0   0   0   0   0   6   0   2   0
    0   0   0   0   0   0   0   0   7   0   1]
 [  0   1   0 587   0   0   1   0   1   0   0   0   0   0   8   0   0   1
    0   0   0   0   0   0   0   0   0   0   1]
 [ 10   3   0   0 570   0   0   0   6   0   0   0   1   0   1   0   0   0
    8   0   0   0   0   0   0   0   0   1   0]
 [  0   0   0   2   0 584   0   0   0   0   0   0  11   0   1   0   0   0
    0   0   0   0   1   0   0   0   0   0   1]
 [  2   2   0   0   1   0 564  10   2   2   0   0   1   0   2   8   0   0
    2   0   1   0   0   0   0   1   0   2   0]
 [  0   2   0   0   0   0   4 581   0   1   0   0   0   0   0   3   1   3
    1   0   2   0   0   0   0   1   0   0   1]
 [  0   2   0   0   3   0   0   0 572   4   3   1   0   1   0   0   0   4
    3   0   4   0   0   0   0   3   0   0   0]
 [  0   0   0   0   1   0   1   4  13 564   0   0   0   0   1   0   0   0
    2   1   0   0   0   0   2  11   0   0   0]
 [  0   2   0   0   0   0   2   0  10   0 571   1   0   0   1   0   0   3
    0   0   1   8   0   1   0   0   0   0   0]
 [  0   0   0   0   0   0   1   0   1   1   0 588   0   0   0   0   0   1
    1   3   0   0   0   0   0   4   0   0   0]
 [ 13   1   0   0   1   0   0   0   0   0   0   0 559  17   0   0   0   0
    5   0   1   0   0   0   0   0   0   0   3]
 [  0   0   0   0   2   0   0   0   0   0   0   0  46 542   4   0   0   0
    3   0   0   0   0   0   0   0   3   0   0]
 [  0   2   2   1   0   0   0   0   0   0   0   0   0   0 591   0   0   0
    0   0   1   0   0   0   0   0   2   1   0]
 [  0   0   2   0   0   0   1   3   0   0   0   0   0   2   2 586   0   0
    1   0   0   0   0   0   0   1   0   1   1]
 [  1   0   0   0   0   0   0   0   0   0   0   0   1   0   0   6 559   0
    0   0   0   0   0   0   1   0  31   0   1]
 [  0   3   0   1   0   0   1   0   0   0   1   0   0   1   1   0   0 530
    1   0  58   0   0   1   0   0   0   1   1]
 [  4   1   0   0   4   0   0   1   1   0   0   0   4   0   2   0   0   1
  557   8   1   0   0  11   0   2   2   0   1]
 [  3   0   0   0   0   0   0   0   0   1   0   1   2   0   0   0   0   0
    6 576   0   0   0   6   4   1   0   0   0]
 [  0   1   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0  17
    3   0 575   0   0   0   0   2   0   1   0]
 [  0   0   0   0   0   0   0   0   0   0  45   0   0   0   0   0   0  14
    0   0   7 520  11   3   0   0   0   0   0]
 [  0   6   0   0   0   1   0   0   0   0   7   0   0   0   0   0   0   7
    0   0   2  19 557   0   0   0   0   0   1]
 [  3   0   0   0   0   0   0   0   3   0   0   3   1   0   1   0   0  10
   19   0   9   0   0 535   0  12   0   0   4]
 [  1   0   0   0   0   0   0   0   0   5   0   2   0   0   0   0   0   0
    1   3   0   0   0   1 577   7   0   0   3]
 [  1   0   0   0   0   0   0   0   0   2   0   0   0   0   0   0   0   0
    5   1   0   0   0   2   1 585   2   0   1]
 [  0   0   0   0   0   0   0   0   0   0   0   0   2   0   1   0   0   0
    0   0   0   0   0   0   0   0 595   0   2]
 [  1   0   0   0   0   0   0   1   0   0   0   0   0   1   0   2   0   0
    1   0   0   0   0   0   1   0   0 588   5]
```

Classification report:

	precision	recall	f1-score	support
A	0.99	0.90	0.94	600
B	0.96	0.94	0.95	600
C	0.99	0.99	0.99	600
D	0.99	0.97	0.98	600
E	0.96	0.90	0.93	600
F	1.00	0.96	0.98	600
G	0.95	0.97	0.96	600
H	0.98	0.94	0.96	600
I	0.96	0.94	0.95	600
J	0.96	0.98	0.97	600
K	0.92	0.95	0.93	600
L	0.99	0.99	0.99	600
M	0.81	0.98	0.88	600
N	0.94	0.87	0.91	600
O	0.96	0.98	0.97	600
P	0.98	0.98	0.98	600
Q	0.97	0.99	0.98	600
R	0.91	0.89	0.90	600
S	0.84	0.94	0.89	600
T	0.99	0.94	0.96	600
U	0.92	0.90	0.91	600
V	0.90	0.89	0.90	600
W	0.97	0.95	0.96	600
X	0.90	0.95	0.92	600
Y	0.96	0.97	0.96	600
Z	0.97	0.96	0.96	600
del	0.98	0.97	0.98	600
nothing	0.99	1.00	0.99	600
space	0.98	0.98	0.98	600
accuracy			0.95	17400
macro avg	0.95	0.95	0.95	17400
weighted avg	0.95	0.95	0.95	17400

Accuracy Score:

| | | | ```
WARNING:absl:`lr` is deprecated in Keras optimizer, please use `learning_rate` or use the legacy optimizer, e.g.,tf.keras.optimizers.legacy.Adam.
Epoch 1/10
543/543 [==============================] - ETA: 0s - loss: 2.8162 - accuracy: 0.1400
Epoch 1: val_accuracy improved from -inf to 0.32923, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 164s 273ms/step - loss: 2.8162 - accuracy: 0.1400 - val_loss: 1.9774 - val_accuracy: 0.3292
Epoch 2/10
543/543 [==============================] - ETA: 0s - loss: 1.2641 - accuracy: 0.5638
Epoch 2: val_accuracy improved from 0.32923 to 0.75509, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 144s 265ms/step - loss: 1.2641 - accuracy: 0.5638 - val_loss: 0.7571 - val_accuracy: 0.7551
Epoch 3/10
543/543 [==============================] - ETA: 0s - loss: 0.5975 - accuracy: 0.8077
Epoch 3: val_accuracy improved from 0.75509 to 0.84929, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 159s 292ms/step - loss: 0.5975 - accuracy: 0.8077 - val_loss: 0.5150 - val_accuracy: 0.8493
Epoch 4/10
543/543 [==============================] - ETA: 0s - loss: 0.3997 - accuracy: 0.8808
Epoch 4: val_accuracy improved from 0.84929 to 0.91094, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 157s 289ms/step - loss: 0.3997 - accuracy: 0.8808 - val_loss: 0.3046 - val_accuracy: 0.9109
Epoch 5/10
543/543 [==============================] - ETA: 0s - loss: 0.3196 - accuracy: 0.9084
Epoch 5: val_accuracy did not improve from 0.91094
543/543 [==============================] - 156s 288ms/step - loss: 0.3196 - accuracy: 0.9084 - val_loss: 0.3662 - val_accuracy: 0.8956
Epoch 6/10
543/543 [==============================] - ETA: 0s - loss: 0.2517 - accuracy: 0.9276
Epoch 6: val_accuracy improved from 0.91094 to 0.94410, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 143s 264ms/step - loss: 0.2517 - accuracy: 0.9276 - val_loss: 0.1956 - val_accuracy: 0.9441
Epoch 7/10
543/543 [==============================] - ETA: 0s - loss: 0.2287 - accuracy: 0.9362
Epoch 7: val_accuracy did not improve from 0.94410
543/543 [==============================] - 145s 267ms/step - loss: 0.2287 - accuracy: 0.9362 - val_loss: 0.2181 - val_accuracy: 0.9393
Epoch 8/10
543/543 [==============================] - ETA: 0s - loss: 0.2239 - accuracy: 0.9378
Epoch 8: val_accuracy did not improve from 0.94410
543/543 [==============================] - 177s 326ms/step - loss: 0.2239 - accuracy: 0.9378 - val_loss: 0.2257 - val_accuracy: 0.9399
Epoch 9/10
543/543 [==============================] - ETA: 0s - loss: 0.2281 - accuracy: 0.9379
Epoch 9: val_accuracy improved from 0.94410 to 0.95282, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 154s 284ms/step - loss: 0.2281 - accuracy: 0.9379 - val_loss: 0.1858 - val_accuracy: 0.9528
Epoch 10/10
543/543 [==============================] - ETA: 0s - loss: 0.2068 - accuracy: 0.9444
Epoch 10: val_accuracy did not improve from 0.95282
543/543 [==============================] - 145s 267ms/step - loss: 0.2068 - accuracy: 0.9444 - val_loss: 0.1917 - val_accuracy: 0.9506
```<br>[13]<br>```
scores = model.evaluate(test_generator)
print("%s: %2f%%" % ("Evaluate Test Accuracy", scores[1]*100))
```<br>```
136/136 [==============================] - 25s 184ms/step - loss: 0.1809 - accuracy: 0.9517
Evaluate Test Accuracy: 95.172411%
```<br>[illegible] |
| **2.** | **Tune the Model** | **Hyperparameter Tuning:**

The model is tuned with following hyper parametersOptimizer - Adam Learning rate - 0.0001 Loss - Categorical cross entropy Batch size - 128 EPOCHS - 10 | **Hyperparameter Tuning:** |
```

		Validation Method: **The validation of the model is done through the validation data , which is set to 20% of training data. Data augmentation and callbacks are also used to validate performance. Accuracy is the validation parameter that we have monitored**	 **Validation Method:**

```
# Configuration
class CFG:
    # Set the batch size for training
    batch_size = 128
    # Set the height and width of input images
    img_height = 32
    img_width = 32
    epochs = 10
    num_classes = 29
    # Define the number of color channels in input images
    img_channels = 3
```

```
# Split the training set into training and validation sets
X_train, X_val, y_train, y_val = train_test_split(
    data_train['image_path'],
    data_train['label'],
    test_size=0.2/0.7,  # Assuming you want 20% for validation out of the training set
    random_state=2253,
    shuffle=True,
    stratify=data_train['label']
)

# Create a DataFrame for the validation set
data_val = pd.DataFrame({
    'image_path': X_val,
    'label': y_val
})
```

```
# Create a ModelCheckpoint callback
checkpoint_callback = ModelCheckpoint(
    filepath='/content/sample_data/best_model_weights.h5',
    monitor='val_accuracy',  # Monitor validation accuracy for saving the best model
    save_best_only=True,
    mode='max',
    verbose=1
)
```

CONFUSION MATRIX:

```
136/136 [==============================] - 25s 177ms/step
tf.Tensor(
[[580   1   1   0   5   0   0   0   0   0   0   0   5   0   0   0   0   0
    8   0   0   0   0   0   0   0   0   0   0]
 [  1 590   0   0   4   1   0   0   1   0   0   0   0   0   1   0   0   0
    0   0   0   0   0   0   0   0   1   1   0]
 [  0   0 583   0   1   0   0   0   0   0   0   0   0   0   6   0   2   0
    0   0   0   0   0   0   0   0   7   0   1]
 [  0   1   0 587   0   0   1   0   1   0   0   0   0   0   8   0   0   1
    0   0   0   0   0   0   0   0   0   0   1]
 [ 10   3   0   0 570   0   0   0   6   0   0   0   1   0   1   0   0   0
    8   0   0   0   0   0   0   0   0   1   0]
 [  0   0   0   2   0 584   0   0   0   0   0   0  11   0   1   0   0   0
    0   0   0   0   1   0   0   0   0   0   1]
 [  2   2   0   0   1   0 564  10   2   2   0   0   1   0   2   8   0   0
    2   0   1   0   0   0   0   1   0   2   0]
 [  0   2   0   0   0   0   4 581   0   1   0   0   0   0   0   3   1   3
    1   0   2   0   0   0   0   1   0   0   1]
 [  0   2   0   0   3   0   0   0 572   4   3   1   0   1   0   0   0   4
    3   0   4   0   0   0   0   3   0   0   0]
 [  0   0   0   0   1   0   1   4  13 564   0   0   0   0   1   0   0   0
    2   1   0   0   0   0   2  11   0   0   0]
 [  0   2   0   0   0   0   2   0  10   0 571   1   0   0   1   0   0   3
    0   0   1   8   0   1   0   0   0   0   0]
 [  0   0   0   0   0   0   1   0   1   1   0 588   0   0   0   0   0   1
    1   3   0   0   0   0   0   4   0   0   0]
 [ 13   1   0   0   1   0   0   0   0   0   0   0 559  17   0   0   0   0
    5   0   1   0   0   0   0   0   0   0   3]
 [  0   0   0   0   2   0   0   0   0   0   0   0  46 542   4   0   0   0
    3   0   0   0   0   0   0   0   3   0   0]
 [  0   2   2   1   0   0   0   0   0   0   0   0   0   0 591   0   0   0
    0   0   1   0   0   0   0   0   2   1   0]
 [  0   0   2   0   0   0   1   3   0   0   0   0   0   2   2 586   0   0
    1   0   0   0   0   0   0   1   0   1   1]
 [  1   0   0   0   0   0   0   0   0   0   0   0   1   0   0   6 559   0
    0   0   0   0   0   0   1   0  31   0   1]
 [  0   3   0   1   0   0   1   0   0   0   1   0   0   1   1   0   0 530
    1   0  58   0   0   1   0   0   0   1   1]
 [  4   1   0   0   4   0   0   1   1   0   0   0   4   0   2   0   0   1
  557   8   1   0   0  11   0   2   2   0   1]
 [  3   0   0   0   0   0   0   0   0   1   0   1   2   0   0   0   0   0
    6 576   0   0   0   6   4   1   0   0   0]
 [  0   1   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0  17
    3   0 575   0   0   0   0   2   0   1   0]
 [  0   0   0   0   0   0   0   0   0   0  45   0   0   0   0   0   0  14
    0   0   7 520  11   3   0   0   0   0   0]
 [  0   6   0   0   0   1   0   0   0   0   7   0   0   0   0   0   0   7
    0   0   2  19 557   0   0   0   0   0   1]
 [  3   0   0   0   0   0   0   0   3   0   0   3   1   0   1   0   0  10
   19   0   9   0   0 535   0  12   0   0   4]
 [  1   0   0   0   0   0   0   0   0   5   0   2   0   0   0   0   0   0
    1   3   0   0   0   1 577   7   0   0   3]
```

CLASSIFICATION REPORT:

	precision	recall	f1-score	support
A	0.99	0.90	0.94	600
B	0.96	0.94	0.95	600
C	0.99	0.99	0.99	600
D	0.99	0.97	0.98	600
E	0.96	0.90	0.93	600
F	1.00	0.96	0.98	600
G	0.95	0.97	0.96	600
H	0.98	0.94	0.96	600
I	0.96	0.94	0.95	600
J	0.96	0.98	0.97	600
K	0.92	0.95	0.93	600
L	0.99	0.99	0.99	600
M	0.81	0.98	0.88	600
N	0.94	0.87	0.91	600
O	0.96	0.98	0.97	600
P	0.98	0.98	0.98	600
Q	0.97	0.99	0.98	600
R	0.91	0.89	0.90	600
S	0.84	0.94	0.89	600
T	0.99	0.94	0.96	600
U	0.92	0.90	0.91	600
V	0.90	0.89	0.90	600
W	0.97	0.95	0.96	600
X	0.90	0.95	0.92	600
Y	0.96	0.97	0.96	600
Z	0.97	0.96	0.96	600
del	0.98	0.97	0.98	600
nothing	0.99	1.00	0.99	600
space	0.98	0.98	0.98	600
accuracy			0.95	17400
macro avg	0.95	0.95	0.95	17400
weighted avg	0.95	0.95	0.95	17400

ACCURACY SCORE:

```
WARNING:absl:`lr` is deprecated in Keras optimizer, please use `learning_rate` or use the legacy optimizer, e.g.,tf.keras.optimizers.legacy.Adam.
Epoch 1/10
543/543 [==============================] - ETA: 0s - loss: 2.8162 - accuracy: 0.1400
Epoch 1: val_accuracy improved from -inf to 0.32923, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 164s 273ms/step - loss: 2.8162 - accuracy: 0.1400 - val_loss: 1.9774 - val_accuracy: 0.3292
Epoch 2/10
543/543 [==============================] - ETA: 0s - loss: 1.2641 - accuracy: 0.5638
Epoch 2: val_accuracy improved from 0.32923 to 0.75509, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 144s 265ms/step - loss: 1.2641 - accuracy: 0.5638 - val_loss: 0.7571 - val_accuracy: 0.7551
Epoch 3/10
543/543 [==============================] - ETA: 0s - loss: 0.5975 - accuracy: 0.8077
Epoch 3: val_accuracy improved from 0.75509 to 0.84929, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 159s 292ms/step - loss: 0.5975 - accuracy: 0.8077 - val_loss: 0.5150 - val_accuracy: 0.8493
Epoch 4/10
543/543 [==============================] - ETA: 0s - loss: 0.3997 - accuracy: 0.8808
Epoch 4: val_accuracy improved from 0.84929 to 0.91094, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 157s 289ms/step - loss: 0.3997 - accuracy: 0.8808 - val_loss: 0.3046 - val_accuracy: 0.9109
Epoch 5/10
543/543 [==============================] - ETA: 0s - loss: 0.3196 - accuracy: 0.9084
Epoch 5: val_accuracy did not improve from 0.91094
543/543 [==============================] - 156s 288ms/step - loss: 0.3196 - accuracy: 0.9084 - val_loss: 0.3662 - val_accuracy: 0.8956
Epoch 6/10
543/543 [==============================] - ETA: 0s - loss: 0.2517 - accuracy: 0.9276
Epoch 6: val_accuracy improved from 0.91094 to 0.94410, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 143s 264ms/step - loss: 0.2517 - accuracy: 0.9276 - val_loss: 0.1956 - val_accuracy: 0.9441
Epoch 7/10
543/543 [==============================] - ETA: 0s - loss: 0.2287 - accuracy: 0.9362
Epoch 7: val_accuracy did not improve from 0.94410
543/543 [==============================] - 145s 267ms/step - loss: 0.2287 - accuracy: 0.9362 - val_loss: 0.2181 - val_accuracy: 0.9393
Epoch 8/10
543/543 [==============================] - ETA: 0s - loss: 0.2239 - accuracy: 0.9378
Epoch 8: val_accuracy did not improve from 0.94410
543/543 [==============================] - 177s 326ms/step - loss: 0.2239 - accuracy: 0.9378 - val_loss: 0.2257 - val_accuracy: 0.9399
Epoch 9/10
543/543 [==============================] - ETA: 0s - loss: 0.2281 - accuracy: 0.9379
Epoch 9: val_accuracy improved from 0.94410 to 0.95282, saving model to /content/sample_data/best_model_weights.h5
543/543 [==============================] - 154s 284ms/step - loss: 0.2281 - accuracy: 0.9379 - val_loss: 0.1858 - val_accuracy: 0.9528
Epoch 10/10
543/543 [==============================] - ETA: 0s - loss: 0.2068 - accuracy: 0.9444
Epoch 10: val_accuracy did not improve from 0.95282
543/543 [==============================] - 145s 267ms/step - loss: 0.2068 - accuracy: 0.9444 - val_loss: 0.1917 - val_accuracy: 0.9506
```

[13]
```
scores = model.evaluate(test_generator)
print("%s: %2f%%" % ("Evaluate Test Accuracy", scores[1]*100))
```

```
136/136 [==============================] - 25s 184ms/step - loss: 0.1809 - accuracy: 0.9517
Evaluate Test Accuracy: 95.172411%
```

HYPERPARAMETER TUNING:

```
# Compile the model
model.compile(optimizer=Adam(lr=0.0001), loss='categorical_crossentropy', metrics=['accuracy'])
```

```
# Configuration
class CFG:
    # Set the batch size for training
    batch_size = 128
    # Set the height and width of input images
    img_height = 32
    img_width = 32
    epochs = 10
    num_classes = 29
```

VALIDATION METHOD:

```
# Split the training set into training and validation sets
X_train, X_val, y_train, y_val = train_test_split(
    data_train['image_path'],
    data_train['label'],
    test_size=0.2/0.7,  # Assuming you want 20% for validation out of the training set
    random_state=2253,
    shuffle=True,
    stratify=data_train['label']
)

# Create a DataFrame for the validation set
data_val = pd.DataFrame({
    'image_path': X_val,
    'label': y_val
})
```

```python
# Create a ModelCheckpoint callback
checkpoint_callback = ModelCheckpoint(
    filepath='/content/sample_data/best_model_weights.h5',
    monitor='val_accuracy',  # Monitor validation accuracy for saving the best model
    save_best_only=True,
    mode='max',
    verbose=1
)
```