

Generating Predictions for Challenge (Test Images)

Create CSV with Predictions

Load best model ...

```
from tensorflow.keras.models import load_model
from tensorflow.keras.layers import Layer
from transformers import TFSViTModel

# Name of model
trial_name = 'baseline'

# Wrapper to convert to Keras layer
class ViTLayer(Layer):
    def __init__(self, vit_model=None, model_name='google/vit-base-patch16-224', **kwargs):
        super(ViTLayer, self).__init__(**kwargs)
        # Load vit_model
        self.vit_model = vit_model if vit_model is not None else
TFSViTModel.from_pretrained(f'{trial_name}_vit_model')
        # Store model name for serialization (needed for
saving/loading)
        self.model_name = model_name

    def call(self, inputs):
        outputs = self.vit_model(inputs)
        return outputs.pooler_output

    def get_config(self):
        config = super(ViTLayer, self).get_config()
        config.update({
            'model_name': self.model_name
        })
        return config

    @classmethod
    def from_config(cls, config):
        # Get model_name and remove it from config to avoid passing to
init
        model_name = config.pop('model_name')
        # Create instance without vit_model (will be loaded in init)
        return cls(model_name=model_name, **config)

# Load model
```

```
model = load_model(f'{trial_name}_transfer_model.keras',  
custom_objects={'ViTLayer': ViTLayer})
```

2025-04-19 00:11:16.215772: E

external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:467] Unable to register cuFFT factory: Attempting to register factory for plugin cuFFT when one has already been registered

WARNING: All log messages before absl::InitializeLog() is called are written to STDERR

E0000 00:00:1745035876.236840 3092294 cuda_dnn.cc:8579] Unable to register cuDNN factory: Attempting to register factory for plugin cuDNN when one has already been registered

E0000 00:00:1745035876.243391 3092294 cuda_blas.cc:1407] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been registered

W0000 00:00:1745035876.259417 3092294 computation_placer.cc:177] computation placer already registered. Please check linkage and avoid linking the same target more than once.

W0000 00:00:1745035876.259430 3092294 computation_placer.cc:177] computation placer already registered. Please check linkage and avoid linking the same target more than once.

W0000 00:00:1745035876.259432 3092294 computation_placer.cc:177] computation placer already registered. Please check linkage and avoid linking the same target more than once.

W0000 00:00:1745035876.259433 3092294 computation_placer.cc:177] computation placer already registered. Please check linkage and avoid linking the same target more than once.

2025-04-19 00:11:16.264474: I

tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.

To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.

/apps/tensorflow/2.18/lib/python3.11/site-packages/tqdm/auto.py:21:

TqdmWarning: IPProgress not found. Please update jupyter and ipywidgets. See

https://ipywidgets.readthedocs.io/en/stable/user_install.html

from .autonotebook import tqdm as notebook_tqdm

2025-04-19 00:11:22.698401: E

external/local_xla/xla/stream_executor/cuda/cuda_platform.cc:51] failed call to cuInit: INTERNAL: CUDA error: Failed call to cuInit: UNKNOWN ERROR (303)

All model checkpoint layers were used when initializing TFCViTModel.

All the layers of TFCViTModel were initialized from the model checkpoint at baseline_vit_model.

If your task is similar to the task the model of the checkpoint was trained on, you can already use TFCViTModel for predictions without further training.

Generate predictions into a CSV ...

```
from general_training import predict_trial
from transformers import AutoImageProcessor
import tensorflow as tf
import numpy as np
from PIL import Image
import pandas as pd
import os

# Load preprocessor
processor = AutoImageProcessor.from_pretrained('google/vit-base-
patch16-224')

# Path to folder with 0001.png ... 10000.png
image_dir = 'competition_test_images/cifar10_test_images'

# Prepare prediction list
predictions = []
image_ids = []

# Loop through 0001.png to 10000.png
for i in range(1, 10001):
    filename = f'{i:04}.png'
    path = os.path.join(image_dir, filename)

    # Load and preprocess image
    img = Image.open(path).resize((32, 32)) # Ensure 32x32
    img_array = np.array(img)
    img_array = np.expand_dims(img_array, axis=0) # Add batch
dimension
    img_array = processor(images=img_array, return_tensors='tf')
['pixel_values']

    # Predict
    pred = model.predict(img_array, verbose=0)
    predicted_class = np.argmax(pred)

    image_ids.append(f'{i:04}')
    predictions.append(predicted_class)

# Save CSV
submission_df = pd.DataFrame({
    'ID': image_ids,
    'PredictedLabel': predictions
})

submission_df.to_csv('MichaelCalderin_section2_submission.csv',
index=False)
print('Submission CSV created!')
```

Submission CSV created!