In [1]:

```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.applications import ResNet50, EfficientNetB0
from tensorflow.keras.optimizers import Adam, SGD
from tensorflow.keras.callbacks import TensorBoard
from tensorflow.keras.models import Model
```

In [2]:

```
n = 4
i1 = n - 1
i2 = n + 29
i3 = n + 59
```

In [3]:

```
data_dir = '101food/train'

classes = sorted(os.listdir(data_dir))
selected_classes = [classes[i1], classes[i2], classes[i3]]
selected_classes
```

Out[3]:

```
['beef_carpaccio', 'edamame', 'macarons']
```

In [4]:

```
train_data = []
test_data = []

for cls in selected_classes:
    cls_path = os.path.join(data_dir, cls)
    cls_images = [os.path.join(cls, img) for img in os.listdir(cls_path)]
    train, test = train_test_split(cls_images, test_size=0.3, random_state=42)
    train_data.extend([(cls, img) for img in train])
    test_data.extend([(cls, img) for img in test])

train_df = pd.DataFrame(train_data, columns=['class', 'image'])
test_df = pd.DataFrame(test_data, columns=['class', 'image'])
```

In [5]:

```
img_size = (224, 224)
batch_size = 32

train_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_dataframe(train_df, directory=data_dir, x_col='image', y_col='class',
test_generator = test_datagen.flow_from_dataframe(test_df, directory=data_dir, x_col='image', y_col='class', ta
```

```
Found 156 validated image filenames belonging to 3 classes.
Found 69 validated image filenames belonging to 3 classes.
```

In [6]:

```
cnn_model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(*img_size, 3)),
    MaxPooling2D(pool_size=(2, 2)),Conv2D(64, (3, 3), activation='relu'),MaxPooling2D(pool_size=(2, 2)),Conv2D(
```

In [7]:

```
def plot_history(model_name, opt_name, history):
    plt.figure(figsize=(12, 4))
    plt.suptitle(f'{model_name} with {opt_name} optimizer')

    plt.subplot(1, 2, 1)
    plt.plot(history.history['accuracy'], label='Train')
    plt.plot(history.history['val_accuracy'], label='Test')
    plt.xlabel('Epoch')
    plt.ylabel('Accuracy')
    plt.legend()

    plt.subplot(1, 2, 2)
    plt.plot(history.history['loss'], label='Train')
    plt.plot(history.history['val_loss'], label='Test')
    plt.xlabel('Epoch')
    plt.ylabel('Loss')
    plt.legend()

    plt.show()
```

In [ ]:

```
from tensorflow.keras.layers import Input

def create_model(model_name, optimizer, img_size=(224, 224)):
    base_model = None
    if model_name == 'resnet':
        base_model = ResNet50(input_shape=(*img_size, 3), include_top=False, weights='imagenet')
    elif model_name == 'efficientnet':
        base_model = EfficientNetB0(input_shape=(*img_size, 3), include_top=False, weights='imagenet')
    for layer in base_model.layers[:-10]:
        layer.trainable = False

    inputs = Input(shape=(*img_size, 3))
    x = base_model(inputs, training=False)
    x = Flatten()(x)
    x = Dense(256, activation='relu', kernel_initializer=HeNormal())(x)
    x = Dropout(0.5)(x)
    outputs = Dense(3, activation='softmax', kernel_initializer=HeNormal())(x)

    model = Model(inputs=inputs, outputs=outputs)
    model.compile(loss='categorical_crossentropy', optimizer=optimizer, metrics=['accuracy'])

    return model
```

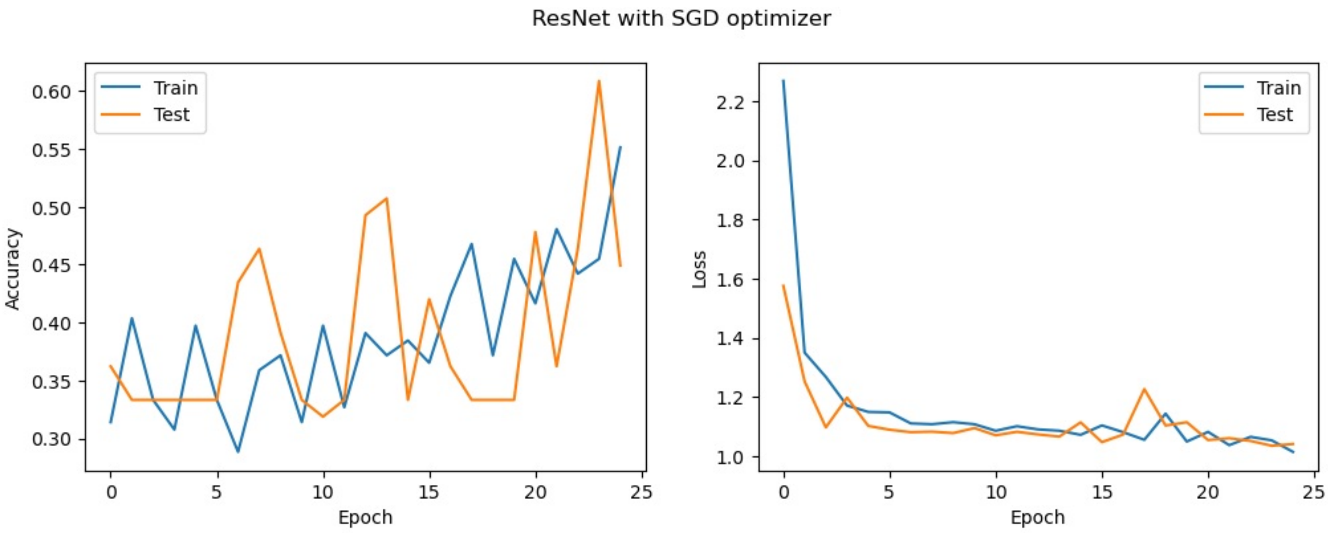
In [ ]:

```
```

In [13]:

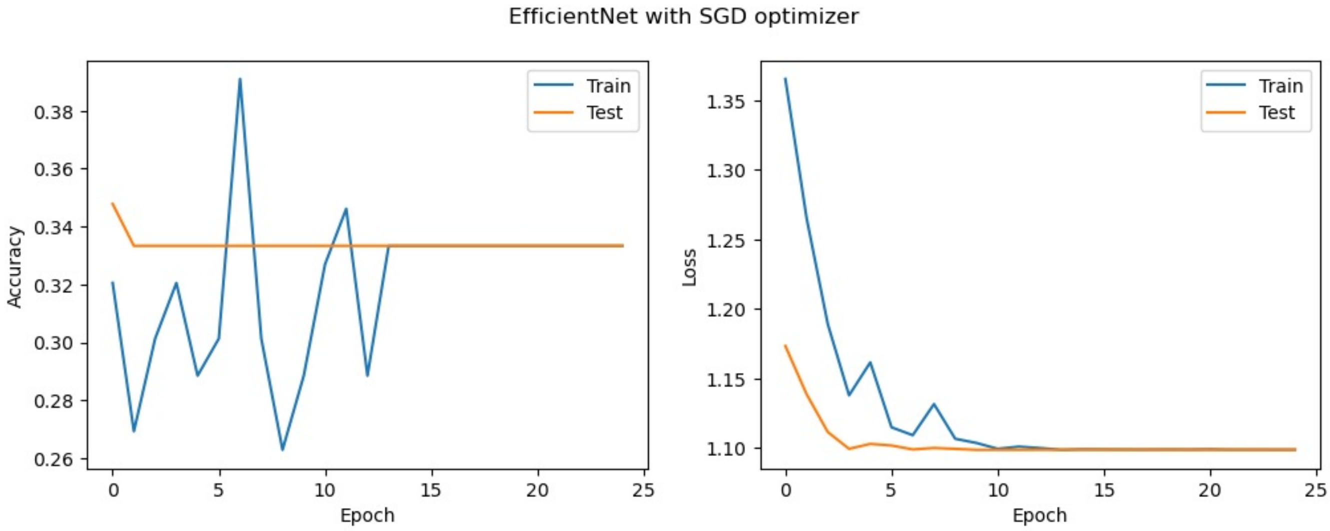
```
from tensorflow.keras.optimizers.schedules import ExponentialDecay

optimizers = {
    'SGD': lambda: SGD(learning_rate=ExponentialDecay(initial_learning_rate=0.01, decay_steps=100000, decay_rat
                       momentum=0.9, nesterov=True, clipnorm=1.0),
    'Adam': lambda: Adam(learning_rate=0.001, clipnorm=1.0),
}
models = {'CNN': 'CNN', 'ResNet': 'resnet', 'EfficientNet': 'efficientnet'}

tensorboard_logs = 'C:\\tensorboard_logs_new'
if not os.path.exists(tensorboard_logs):
    os.makedirs(tensorboard_logs)

for opt_name, opt_func in optimizers.items():
    for model_name, model_type in models.items():
        if model_name == 'CNN' and opt_name == 'SGD':
            continue

        log_dir = os.path.join(tensorboard_logs, f'{model_name}_{opt_name}')
        tensorboard_callback = TensorBoard(log_dir=log_dir, histogram_freq=1)

        print(f'Training {model_name} with {opt_name} optimizer')

        opt = opt_func()

        if model_name == 'CNN':
            model = cnn_model
            model.compile(loss='categorical_crossentropy', optimizer=opt, metrics=['accuracy'])

            opt.build(model.trainable_variables)
        else:
            model = create_model(model_type, opt)

        history = model.fit(train_generator, epochs=25, validation_data=test_generator, callbacks=[tensorboard_

        plot_history(model_name, opt_name, history)
```

```
Training ResNet with SGD optimizer
Epoch 1/25
5/5 [==============================] - 13s 2s/step - loss: 2.2687 - accuracy: 0.3141 - val_loss: 1.5757 - val_a
ccuracy: 0.3623
Epoch 2/25
5/5 [==============================] - 11s 2s/step - loss: 1.3499 - accuracy: 0.4038 - val_loss: 1.2527 - val_a
ccuracy: 0.3333
Epoch 3/25
5/5 [==============================] - 11s 2s/step - loss: 1.2675 - accuracy: 0.3333 - val_loss: 1.0969 - val_a
ccuracy: 0.3333
Epoch 4/25
5/5 [==============================] - 11s 2s/step - loss: 1.1708 - accuracy: 0.3077 - val_loss: 1.1974 - val_a
ccuracy: 0.3333
Epoch 5/25
5/5 [==============================] - 11s 2s/step - loss: 1.1493 - accuracy: 0.3974 - val_loss: 1.1020 - val_a
ccuracy: 0.3333
Epoch 6/25
5/5 [==============================] - 11s 2s/step - loss: 1.1477 - accuracy: 0.3333 - val_loss: 1.0891 - val_a
ccuracy: 0.3333
Epoch 7/25
5/5 [==============================] - 11s 2s/step - loss: 1.1104 - accuracy: 0.2885 - val_loss: 1.0808 - val_a
ccuracy: 0.4348
Epoch 8/25
5/5 [==============================] - 11s 2s/step - loss: 1.1077 - accuracy: 0.3590 - val_loss: 1.0825 - val_a
ccuracy: 0.4638
Epoch 9/25
5/5 [==============================] - 11s 2s/step - loss: 1.1146 - accuracy: 0.3718 - val_loss: 1.0778 - val_a
ccuracy: 0.3913
Epoch 10/25
5/5 [==============================] - 11s 2s/step - loss: 1.1076 - accuracy: 0.3141 - val_loss: 1.0946 - val_a
ccuracy: 0.3333
Epoch 11/25
5/5 [==============================] - 11s 2s/step - loss: 1.0856 - accuracy: 0.3974 - val_loss: 1.0703 - val_a
ccuracy: 0.3188
Epoch 12/25
5/5 [==============================] - 11s 2s/step - loss: 1.1009 - accuracy: 0.3269 - val_loss: 1.0821 - val_a
ccuracy: 0.3333
Epoch 13/25
5/5 [==============================] - 11s 2s/step - loss: 1.0901 - accuracy: 0.3910 - val_loss: 1.0731 - val_a
ccuracy: 0.4928
Epoch 14/25
5/5 [==============================] - 11s 2s/step - loss: 1.0857 - accuracy: 0.3718 - val_loss: 1.0659 - val_a
ccuracy: 0.5072
Epoch 15/25
5/5 [==============================] - 11s 2s/step - loss: 1.0718 - accuracy: 0.3846 - val_loss: 1.1141 - val_a
ccuracy: 0.3333
Epoch 16/25
5/5 [==============================] - 11s 2s/step - loss: 1.1036 - accuracy: 0.3654 - val_loss: 1.0471 - val_a
ccuracy: 0.4203
Epoch 17/25
5/5 [==============================] - 11s 2s/step - loss: 1.0809 - accuracy: 0.4231 - val_loss: 1.0727 - val_a
ccuracy: 0.3623
Epoch 18/25
5/5 [==============================] - 11s 2s/step - loss: 1.0550 - accuracy: 0.4679 - val_loss: 1.2265 - val_a
ccuracy: 0.3333
Epoch 19/25
5/5 [==============================] - 11s 2s/step - loss: 1.1436 - accuracy: 0.3718 - val_loss: 1.1033 - val_a
ccuracy: 0.3333
Epoch 20/25
5/5 [==============================] - 11s 2s/step - loss: 1.0491 - accuracy: 0.4551 - val_loss: 1.1144 - val_a
ccuracy: 0.3333
Epoch 21/25
5/5 [==============================] - 11s 2s/step - loss: 1.0818 - accuracy: 0.4167 - val_loss: 1.0542 - val_a
ccuracy: 0.4783
Epoch 22/25
5/5 [==============================] - 11s 2s/step - loss: 1.0369 - accuracy: 0.4808 - val_loss: 1.0604 - val_a
ccuracy: 0.3623
Epoch 23/25
5/5 [==============================] - 11s 2s/step - loss: 1.0650 - accuracy: 0.4423 - val_loss: 1.0512 - val_a
ccuracy: 0.4638
Epoch 24/25
5/5 [==============================] - 11s 2s/step - loss: 1.0531 - accuracy: 0.4551 - val_loss: 1.0349 - val_a
ccuracy: 0.6087
Epoch 25/25
5/5 [==============================] - 11s 2s/step - loss: 1.0142 - accuracy: 0.5513 - val_loss: 1.0408 - val_a
ccuracy: 0.4493
```

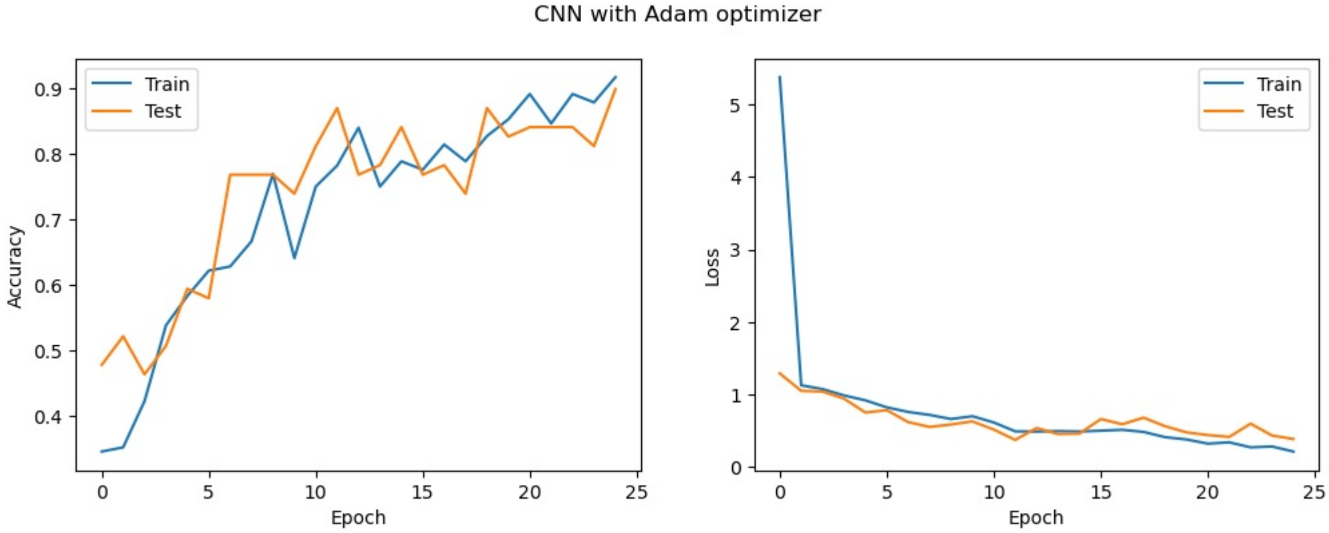


ResNet with SGD optimizer

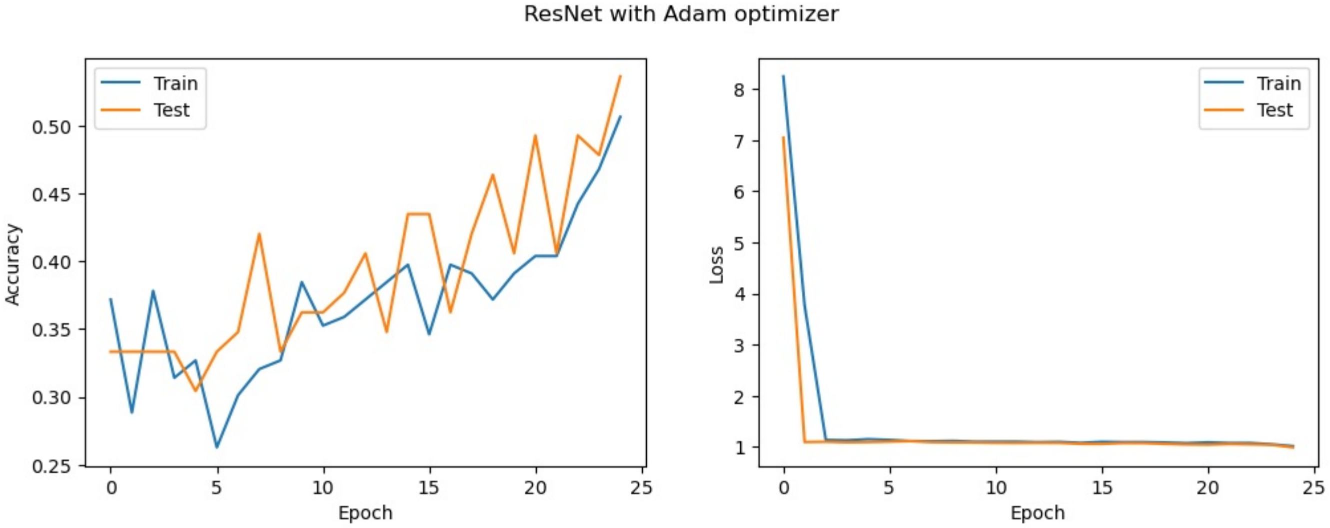
```
Training EfficientNet with SGD optimizer
WARNING:tensorflow:Model failed to serialize as JSON. Ignoring... Unable to serialize [2.0896919 2.1128857 2.10
81853] to JSON. Unrecognized type <class 'tensorflow.python.framework.ops.EagerTensor'>.
Epoch 1/25
5/5 [==============================] - 10s 2s/step - loss: 1.3655 - accuracy: 0.3205 - val_loss: 1.1733 - val_a
ccuracy: 0.3478
Epoch 2/25
5/5 [==============================] - 6s 1s/step - loss: 1.2653 - accuracy: 0.2692 - val_loss: 1.1387 - val_ac
curacy: 0.3333
Epoch 3/25
5/5 [==============================] - 6s 1s/step - loss: 1.1890 - accuracy: 0.3013 - val_loss: 1.1114 - val_ac
curacy: 0.3333
Epoch 4/25
5/5 [==============================] - 6s 1s/step - loss: 1.1378 - accuracy: 0.3205 - val_loss: 1.0994 - val_ac
curacy: 0.3333
Epoch 5/25
5/5 [==============================] - 6s 1s/step - loss: 1.1616 - accuracy: 0.2885 - val_loss: 1.1029 - val_ac
curacy: 0.3333
Epoch 6/25
5/5 [==============================] - 6s 1s/step - loss: 1.1148 - accuracy: 0.3013 - val_loss: 1.1018 - val_ac
curacy: 0.3333
Epoch 7/25
5/5 [==============================] - 6s 1s/step - loss: 1.1092 - accuracy: 0.3910 - val_loss: 1.0989 - val_ac
curacy: 0.3333
Epoch 8/25
5/5 [==============================] - 6s 1s/step - loss: 1.1316 - accuracy: 0.3013 - val_loss: 1.1000 - val_ac
curacy: 0.3333
Epoch 9/25
5/5 [==============================] - 6s 1s/step - loss: 1.1066 - accuracy: 0.2628 - val_loss: 1.0993 - val_ac
curacy: 0.3333
Epoch 10/25
5/5 [==============================] - 6s 1s/step - loss: 1.1036 - accuracy: 0.2885 - val_loss: 1.0986 - val_ac
curacy: 0.3333
Epoch 11/25
5/5 [==============================] - 6s 1s/step - loss: 1.0994 - accuracy: 0.3269 - val_loss: 1.0986 - val_ac
curacy: 0.3333
Epoch 12/25
5/5 [==============================] - 7s 1s/step - loss: 1.1010 - accuracy: 0.3462 - val_loss: 1.0986 - val_ac
curacy: 0.3333
Epoch 13/25
5/5 [==============================] - 6s 1s/step - loss: 1.0999 - accuracy: 0.2885 - val_loss: 1.0987 - val_ac
curacy: 0.3333
Epoch 14/25
5/5 [==============================] - 7s 1s/step - loss: 1.0987 - accuracy: 0.3333 - val_loss: 1.0987 - val_ac
curacy: 0.3333
Epoch 15/25
5/5 [==============================] - 7s 1s/step - loss: 1.0990 - accuracy: 0.3333 - val_loss: 1.0987 - val_ac
curacy: 0.3333
Epoch 16/25
5/5 [==============================] - 7s 1s/step - loss: 1.0989 - accuracy: 0.3333 - val_loss: 1.0987 - val_ac
curacy: 0.3333
Epoch 17/25
5/5 [==============================] - 7s 1s/step - loss: 1.0989 - accuracy: 0.3333 - val_loss: 1.0987 - val_ac
curacy: 0.3333
Epoch 18/25
5/5 [==============================] - 6s 1s/step - loss: 1.0988 - accuracy: 0.3333 - val_loss: 1.0987 - val_ac
curacy: 0.3333
Epoch 19/25
5/5 [==============================] - 6s 1s/step - loss: 1.0989 - accuracy: 0.3333 - val_loss: 1.0986 - val_ac
curacy: 0.3333
Epoch 20/25
5/5 [==============================] - 6s 1s/step - loss: 1.0989 - accuracy: 0.3333 - val_loss: 1.0986 - val_ac
curacy: 0.3333
Epoch 21/25
5/5 [==============================] - 6s 1s/step - loss: 1.0990 - accuracy: 0.3333 - val_loss: 1.0987 - val_ac
curacy: 0.3333
Epoch 22/25
5/5 [==============================] - 7s 1s/step - loss: 1.0988 - accuracy: 0.3333 - val_loss: 1.0986 - val_ac
curacy: 0.3333
Epoch 23/25
5/5 [==============================] - 7s 2s/step - loss: 1.0988 - accuracy: 0.3333 - val_loss: 1.0986 - val_ac
curacy: 0.3333
Epoch 24/25
5/5 [==============================] - 7s 1s/step - loss: 1.0988 - accuracy: 0.3333 - val_loss: 1.0986 - val_ac
curacy: 0.3333
Epoch 25/25
5/5 [==============================] - 7s 1s/step - loss: 1.0988 - accuracy: 0.3333 - val_loss: 1.0986 - val_ac
curacy: 0.3333
```


EfficientNet with SGD optimizer
Train
Test
Accuracy
0.38
0.36
0.34
0.32
0.30
0.28
0.26
0
5
10
15
20
25
Epoch
Train
Test
Loss
1.35
1.30
1.25
1.20
1.15
1.10
0
5
10
15
20
25
Epoch

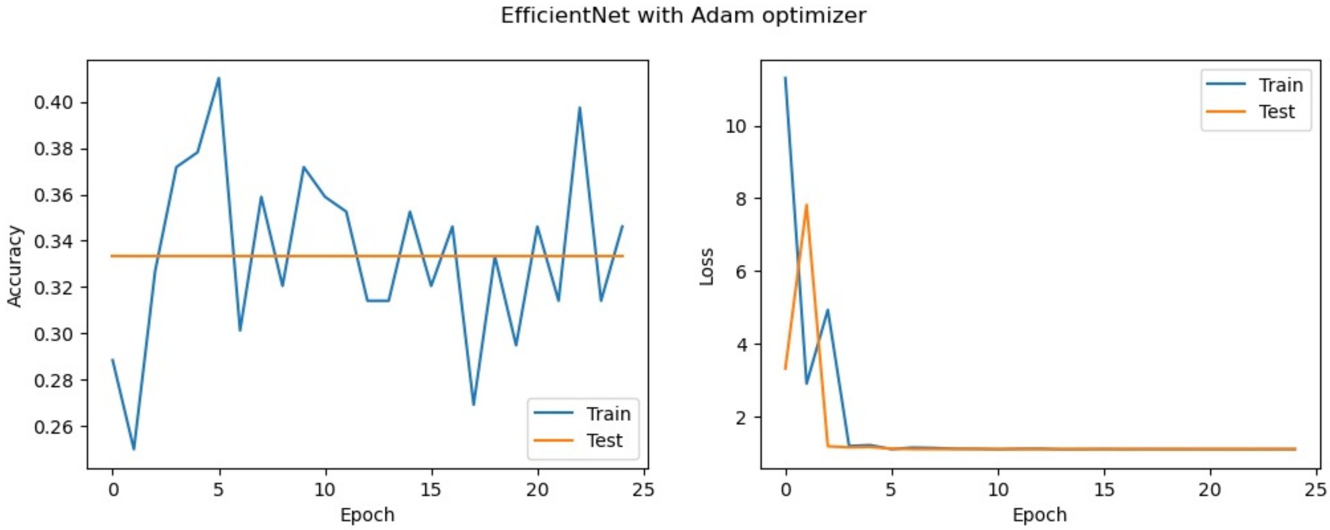
```
Training CNN with Adam optimizer
Epoch 1/25
5/5 [==============================] - 6s 1s/step - loss: 5.3781 - accuracy: 0.3462 - val_loss: 1.2913 - val_ac
curacy: 0.4783
Epoch 2/25
5/5 [==============================] - 5s 1s/step - loss: 1.1281 - accuracy: 0.3526 - val_loss: 1.0513 - val_ac
curacy: 0.5217
Epoch 3/25
5/5 [==============================] - 5s 1s/step - loss: 1.0750 - accuracy: 0.4231 - val_loss: 1.0414 - val_ac
curacy: 0.4638
Epoch 4/25
5/5 [==============================] - 5s 1s/step - loss: 0.9884 - accuracy: 0.5385 - val_loss: 0.9435 - val_ac
curacy: 0.5072
Epoch 5/25
5/5 [==============================] - 5s 1s/step - loss: 0.9203 - accuracy: 0.5833 - val_loss: 0.7524 - val_ac
curacy: 0.5942
Epoch 6/25
5/5 [==============================] - 6s 1s/step - loss: 0.8233 - accuracy: 0.6218 - val_loss: 0.7844 - val_ac
curacy: 0.5797
Epoch 7/25
5/5 [==============================] - 6s 1s/step - loss: 0.7600 - accuracy: 0.6282 - val_loss: 0.6203 - val_ac
curacy: 0.7681
Epoch 8/25
5/5 [==============================] - 6s 1s/step - loss: 0.7195 - accuracy: 0.6667 - val_loss: 0.5533 - val_ac
curacy: 0.7681
Epoch 9/25
5/5 [==============================] - 5s 1s/step - loss: 0.6628 - accuracy: 0.7692 - val_loss: 0.5869 - val_ac
curacy: 0.7681
Epoch 10/25
5/5 [==============================] - 5s 1s/step - loss: 0.7010 - accuracy: 0.6410 - val_loss: 0.6299 - val_ac
curacy: 0.7391
Epoch 11/25
5/5 [==============================] - 5s 1s/step - loss: 0.6147 - accuracy: 0.7500 - val_loss: 0.5168 - val_ac
curacy: 0.8116
Epoch 12/25
5/5 [==============================] - 5s 1s/step - loss: 0.4926 - accuracy: 0.7821 - val_loss: 0.3749 - val_ac
curacy: 0.8696
Epoch 13/25
5/5 [==============================] - 6s 1s/step - loss: 0.4896 - accuracy: 0.8397 - val_loss: 0.5376 - val_ac
curacy: 0.7681
Epoch 14/25
5/5 [==============================] - 5s 1s/step - loss: 0.4957 - accuracy: 0.7500 - val_loss: 0.4561 - val_ac
curacy: 0.7826
Epoch 15/25
5/5 [==============================] - 5s 1s/step - loss: 0.4902 - accuracy: 0.7885 - val_loss: 0.4613 - val_ac
curacy: 0.8406
Epoch 16/25
5/5 [==============================] - 5s 1s/step - loss: 0.5018 - accuracy: 0.7756 - val_loss: 0.6621 - val_ac
curacy: 0.7681
Epoch 17/25
5/5 [==============================] - 5s 1s/step - loss: 0.5138 - accuracy: 0.8141 - val_loss: 0.5905 - val_ac
curacy: 0.7826
Epoch 18/25
5/5 [==============================] - 5s 1s/step - loss: 0.4846 - accuracy: 0.7885 - val_loss: 0.6811 - val_ac
curacy: 0.7391
Epoch 19/25
5/5 [==============================] - 5s 1s/step - loss: 0.4124 - accuracy: 0.8269 - val_loss: 0.5637 - val_ac
curacy: 0.8696
Epoch 20/25
5/5 [==============================] - 5s 1s/step - loss: 0.3808 - accuracy: 0.8526 - val_loss: 0.4785 - val_ac
curacy: 0.8261
Epoch 21/25
5/5 [==============================] - 5s 1s/step - loss: 0.3230 - accuracy: 0.8910 - val_loss: 0.4412 - val_ac
curacy: 0.8406
Epoch 22/25
5/5 [==============================] - 5s 1s/step - loss: 0.3411 - accuracy: 0.8462 - val_loss: 0.4164 - val_ac
curacy: 0.8406
Epoch 23/25
5/5 [==============================] - 5s 1s/step - loss: 0.2720 - accuracy: 0.8910 - val_loss: 0.5983 - val_ac
curacy: 0.8406
Epoch 24/25
5/5 [==============================] - 5s 1s/step - loss: 0.2833 - accuracy: 0.8782 - val_loss: 0.4350 - val_ac
curacy: 0.8116
Epoch 25/25
5/5 [==============================] - 5s 1s/step - loss: 0.2138 - accuracy: 0.9167 - val_loss: 0.3867 - val_ac
curacy: 0.8986
```


CNN with Adam optimizer
Train
Test
Accuracy
0.4
0.5
0.6
0.7
0.8
0.9
Epoch
0
5
10
15
20
25
Train
Test
Loss
0
1
2
3
4
5
Epoch
0
5
10
15
20
25

```
Training ResNet with Adam optimizer
Epoch 1/25
5/5 [==============================] - 15s 3s/step - loss: 8.2445 - accuracy: 0.3718 - val_loss: 7.0467 - val_a
ccuracy: 0.3333
Epoch 2/25
5/5 [==============================] - 11s 2s/step - loss: 3.7663 - accuracy: 0.2885 - val_loss: 1.0970 - val_a
ccuracy: 0.3333
Epoch 3/25
5/5 [==============================] - 11s 2s/step - loss: 1.1381 - accuracy: 0.3782 - val_loss: 1.1026 - val_a
ccuracy: 0.3333
Epoch 4/25
5/5 [==============================] - 11s 2s/step - loss: 1.1309 - accuracy: 0.3141 - val_loss: 1.0908 - val_a
ccuracy: 0.3333
Epoch 5/25
5/5 [==============================] - 11s 2s/step - loss: 1.1533 - accuracy: 0.3269 - val_loss: 1.0958 - val_a
ccuracy: 0.3043
Epoch 6/25
5/5 [==============================] - 11s 3s/step - loss: 1.1388 - accuracy: 0.2628 - val_loss: 1.1050 - val_a
ccuracy: 0.3333
Epoch 7/25
5/5 [==============================] - 11s 3s/step - loss: 1.1161 - accuracy: 0.3013 - val_loss: 1.1155 - val_a
ccuracy: 0.3478
Epoch 8/25
5/5 [==============================] - 11s 3s/step - loss: 1.1137 - accuracy: 0.3205 - val_loss: 1.0946 - val_a
ccuracy: 0.4203
Epoch 9/25
5/5 [==============================] - 11s 2s/step - loss: 1.1196 - accuracy: 0.3269 - val_loss: 1.0894 - val_a
ccuracy: 0.3333
Epoch 10/25
5/5 [==============================] - 11s 2s/step - loss: 1.1046 - accuracy: 0.3846 - val_loss: 1.0883 - val_a
ccuracy: 0.3623
Epoch 11/25
5/5 [==============================] - 11s 2s/step - loss: 1.1057 - accuracy: 0.3526 - val_loss: 1.0823 - val_a
ccuracy: 0.3623
Epoch 12/25
5/5 [==============================] - 11s 2s/step - loss: 1.1055 - accuracy: 0.3590 - val_loss: 1.0803 - val_a
ccuracy: 0.3768
Epoch 13/25
5/5 [==============================] - 11s 2s/step - loss: 1.0954 - accuracy: 0.3718 - val_loss: 1.0824 - val_a
ccuracy: 0.4058
Epoch 14/25
5/5 [==============================] - 11s 3s/step - loss: 1.1012 - accuracy: 0.3846 - val_loss: 1.0809 - val_a
ccuracy: 0.3478
Epoch 15/25
5/5 [==============================] - 11s 2s/step - loss: 1.0803 - accuracy: 0.3974 - val_loss: 1.0618 - val_a
ccuracy: 0.4348
Epoch 16/25
5/5 [==============================] - 11s 2s/step - loss: 1.1013 - accuracy: 0.3462 - val_loss: 1.0607 - val_a
ccuracy: 0.4348
Epoch 17/25
5/5 [==============================] - 11s 3s/step - loss: 1.0960 - accuracy: 0.3974 - val_loss: 1.0772 - val_a
ccuracy: 0.3623
Epoch 18/25
5/5 [==============================] - 12s 3s/step - loss: 1.0964 - accuracy: 0.3910 - val_loss: 1.0755 - val_a
ccuracy: 0.4203
Epoch 19/25
5/5 [==============================] - 11s 2s/step - loss: 1.0877 - accuracy: 0.3718 - val_loss: 1.0610 - val_a
ccuracy: 0.4638
Epoch 20/25
5/5 [==============================] - 12s 3s/step - loss: 1.0760 - accuracy: 0.3910 - val_loss: 1.0499 - val_a
ccuracy: 0.4058
Epoch 21/25
5/5 [==============================] - 11s 2s/step - loss: 1.0909 - accuracy: 0.4038 - val_loss: 1.0459 - val_a
ccuracy: 0.4928
Epoch 22/25
5/5 [==============================] - 11s 2s/step - loss: 1.0782 - accuracy: 0.4038 - val_loss: 1.0584 - val_a
ccuracy: 0.4058
Epoch 23/25
5/5 [==============================] - 12s 3s/step - loss: 1.0786 - accuracy: 0.4423 - val_loss: 1.0521 - val_a
ccuracy: 0.4928
Epoch 24/25
5/5 [==============================] - 12s 3s/step - loss: 1.0535 - accuracy: 0.4679 - val_loss: 1.0431 - val_a
ccuracy: 0.4783
Epoch 25/25
5/5 [==============================] - 12s 3s/step - loss: 1.0174 - accuracy: 0.5064 - val_loss: 0.9931 - val_a
ccuracy: 0.5362
```


ResNet with Adam optimizer
Accuracy
0.25
0.30
0.35
0.40
0.45
0.50
Epoch
0
5
10
15
20
25
Train
Test
Loss
1
2
3
4
5
6
7
8
Epoch
0
5
10
15
20
25
Train
Test

```
Training EfficientNet with Adam optimizer
WARNING:tensorflow:Model failed to serialize as JSON. Ignoring... Unable to serialize [2.0896919 2.1128857 2.10
81853] to JSON. Unrecognized type <class 'tensorflow.python.framework.ops.EagerTensor'>.
Epoch 1/25
5/5 [==============================] - 12s 2s/step - loss: 11.3080 - accuracy: 0.2885 - val_loss: 3.3150 - val_
accuracy: 0.3333
Epoch 2/25
5/5 [==============================] - 7s 1s/step - loss: 2.9007 - accuracy: 0.2500 - val_loss: 7.8186 - val_ac
curacy: 0.3333
Epoch 3/25
5/5 [==============================] - 7s 2s/step - loss: 4.9319 - accuracy: 0.3269 - val_loss: 1.1779 - val_ac
curacy: 0.3333
Epoch 4/25
5/5 [==============================] - 7s 1s/step - loss: 1.1876 - accuracy: 0.3718 - val_loss: 1.1500 - val_ac
curacy: 0.3333
Epoch 5/25
5/5 [==============================] - 7s 1s/step - loss: 1.2101 - accuracy: 0.3782 - val_loss: 1.1608 - val_ac
curacy: 0.3333
Epoch 6/25
5/5 [==============================] - 7s 1s/step - loss: 1.1001 - accuracy: 0.4103 - val_loss: 1.1122 - val_ac
curacy: 0.3333
Epoch 7/25
5/5 [==============================] - 6s 1s/step - loss: 1.1428 - accuracy: 0.3013 - val_loss: 1.1023 - val_ac
curacy: 0.3333
Epoch 8/25
5/5 [==============================] - 6s 1s/step - loss: 1.1312 - accuracy: 0.3590 - val_loss: 1.0996 - val_ac
curacy: 0.3333
Epoch 9/25
5/5 [==============================] - 7s 1s/step - loss: 1.1116 - accuracy: 0.3205 - val_loss: 1.1020 - val_ac
curacy: 0.3333
Epoch 10/25
5/5 [==============================] - 7s 1s/step - loss: 1.1092 - accuracy: 0.3718 - val_loss: 1.1028 - val_ac
curacy: 0.3333
Epoch 11/25
5/5 [==============================] - 6s 1s/step - loss: 1.0973 - accuracy: 0.3590 - val_loss: 1.1005 - val_ac
curacy: 0.3333
Epoch 12/25
5/5 [==============================] - 6s 1s/step - loss: 1.1085 - accuracy: 0.3526 - val_loss: 1.0993 - val_ac
curacy: 0.3333
Epoch 13/25
5/5 [==============================] - 6s 1s/step - loss: 1.1114 - accuracy: 0.3141 - val_loss: 1.0988 - val_ac
curacy: 0.3333
Epoch 14/25
5/5 [==============================] - 7s 1s/step - loss: 1.0964 - accuracy: 0.3141 - val_loss: 1.0995 - val_ac
curacy: 0.3333
Epoch 15/25
5/5 [==============================] - 6s 1s/step - loss: 1.0989 - accuracy: 0.3526 - val_loss: 1.0990 - val_ac
curacy: 0.3333
Epoch 16/25
5/5 [==============================] - 6s 1s/step - loss: 1.1029 - accuracy: 0.3205 - val_loss: 1.0989 - val_ac
curacy: 0.3333
Epoch 17/25
5/5 [==============================] - 6s 1s/step - loss: 1.0989 - accuracy: 0.3462 - val_loss: 1.0988 - val_ac
curacy: 0.3333
Epoch 18/25
5/5 [==============================] - 7s 1s/step - loss: 1.1012 - accuracy: 0.2692 - val_loss: 1.0987 - val_ac
curacy: 0.3333
Epoch 19/25
5/5 [==============================] - 7s 1s/step - loss: 1.0992 - accuracy: 0.3333 - val_loss: 1.0987 - val_ac
curacy: 0.3333
Epoch 20/25
5/5 [==============================] - 7s 1s/step - loss: 1.0998 - accuracy: 0.2949 - val_loss: 1.0987 - val_ac
curacy: 0.3333
Epoch 21/25
5/5 [==============================] - 7s 1s/step - loss: 1.0985 - accuracy: 0.3462 - val_loss: 1.0987 - val_ac
curacy: 0.3333
Epoch 22/25
5/5 [==============================] - 7s 1s/step - loss: 1.0993 - accuracy: 0.3141 - val_loss: 1.0986 - val_ac
curacy: 0.3333
Epoch 23/25
5/5 [==============================] - 7s 1s/step - loss: 1.0979 - accuracy: 0.3974 - val_loss: 1.0987 - val_ac
curacy: 0.3333
Epoch 24/25
5/5 [==============================] - 6s 1s/step - loss: 1.0999 - accuracy: 0.3141 - val_loss: 1.0987 - val_ac
curacy: 0.3333
Epoch 25/25
5/5 [==============================] - 6s 1s/step - loss: 1.0985 - accuracy: 0.3462 - val_loss: 1.0987 - val_ac
curacy: 0.3333
```



In [ ]:

In [ ]:

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js