Pistachio Classification & Fine Tuning With EfficientNet

About Pistachio

The pistachio is a member of the cashew family, is a small tree originating from Central Asia and the Middle East. The tree produces seeds that are widely consumed as food.

Inroduction

In this notebook, we will learn how to classify images of Pistachio by making a fine tuning to a pre-trained network.

what we will learn.

Load the data of images.

Visulaise the Data distribution of all data

Visulaise the Data distribution of train and test data

Visualizing some of the images

Efficient NetB7

Fine Tuning

Setting up callbacks

Graph the training loss and validation loss

Classification Report

Prediction Comparison

Confusion Matrix

Importing Libraries

In [2]:
```
!pip install split-folders
```

```
Collecting split-folders
  Downloading split_folders-0.5.1-py3-none-any.whl (8.4 kB)
Installing collected packages: split-folders
Successfully installed split-folders-0.5.1
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
```

In [3]:
```
import matplotlib.pyplot as plt
import numpy as np
```

```
import os
import PIL
import tensorflow as tf
import pathlib
import cv2
from keras.preprocessing.image import ImageDataGenerator
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential
from keras.models import Sequential, Model,load_model
from keras.callbacks import EarlyStopping,ModelCheckpoint
from keras.layers import Input, Add, Dense, Activation, ZeroPadding2D, BatchNormalization, Flatten, Conv2D, AveragePooling2D, MaxPooling2D, GlobalMaxPooling2D,MaxPool2D
from keras.preprocessing import image
from keras.initializers import glorot_uniform
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, cohen_kappa_score, roc_auc_score, confusion_matrix
from sklearn.metrics import classification_report
from keras.layers import Input, Add, Dense, Activation, ZeroPadding2D, BatchNormalization, Flatten, Conv2D, AveragePooling2D, MaxPooling2D, GlobalMaxPooling2D,MaxPool2D,Dropout
import tensorflow as tf
import splitfolders
import pandas as pd
import glob
from sklearn.metrics import confusion_matrix
import plotly.graph_objects as go
import itertools
import plotly.express as px
#Suppressing Warnings
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
```

Importing Data

In [4]:
```
data_dir = "../input/Pistachio_Image_Dataset/Pistachio_Image_Dataset/"
data_dir =pathlib.Path(data_dir)
```

In [5]:
```
Total_Images = glob.glob('../input/Pistachio_Image_Dataset/Pistachio_Image_Dataset/*/*.jpg')
print("Total number of images: ", len(Total_Images))

Total_Images = pd.Series(Total_Images)
```

```
Total number of images:  2148
```

In [6]:
```
total_df = pd.DataFrame()

# generate Filename field
total_df['Filename'] = Total_Images.map( lambda img_name: img_name.split("/")[-1])


# generate ClassId field
total_df['ClassId'] = Total_Images.map(lambda img_name: img_name.split("/")[-2])

total_df.head()
```

Out[6]:

	Filename	ClassId
0	kirmizi 1109.jpg	Kirmizi_Pistachio
1	kirmizi (10).jpg	Kirmizi_Pistachio
2	kirmizi 97.jpg	Kirmizi_Pistachio
3	kirmizi 475.jpg	Kirmizi_Pistachio
4	kirmizi 29.jpg	Kirmizi_Pistachio

In [7]:
```
class_id_distributionTotal = total_df['ClassId'].value_counts()
class_id_distributionTotal.head(10)
```

Out[7]:
```
Kirmizi_Pistachio    1232
Siirt_Pistachio       916
Name: ClassId, dtype: int64
```

Total Distribution of Data

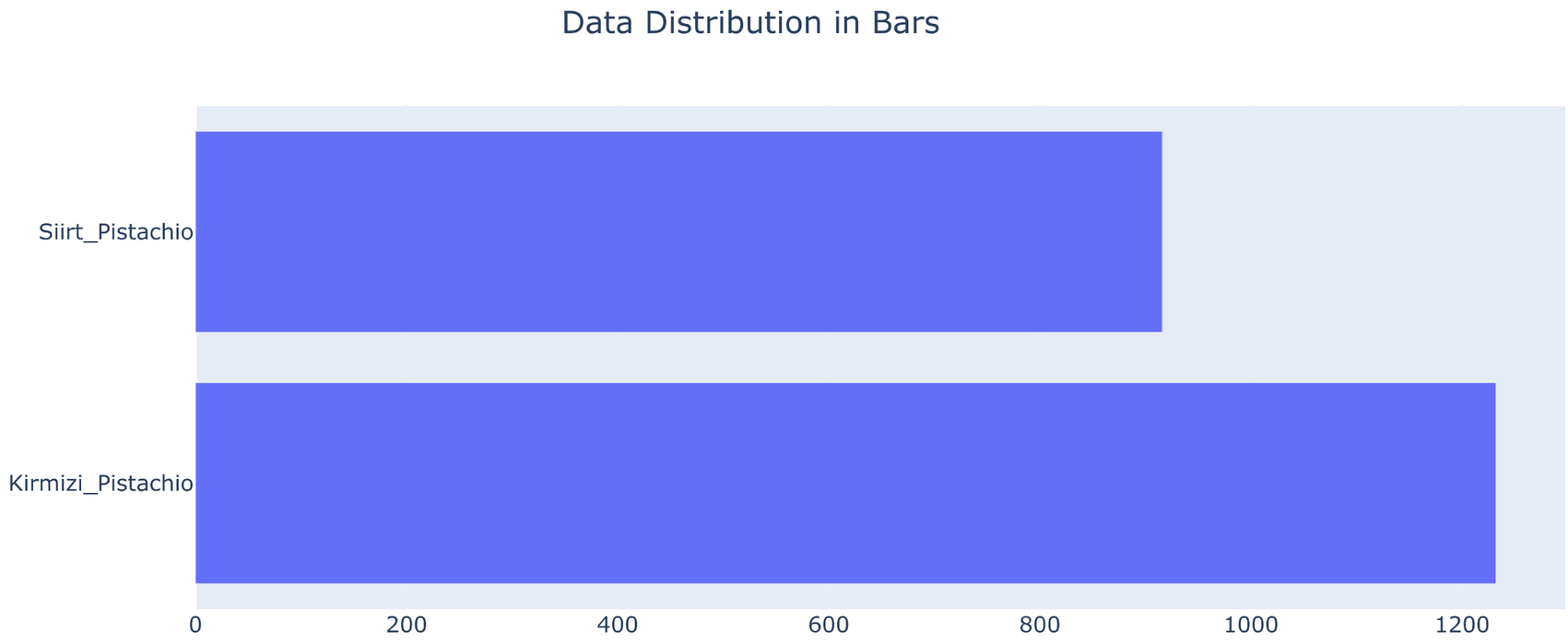
Below charts shows the perentage of data of two different classes.

"Kirmizi_Pistachio " class contains 57.4% from data.

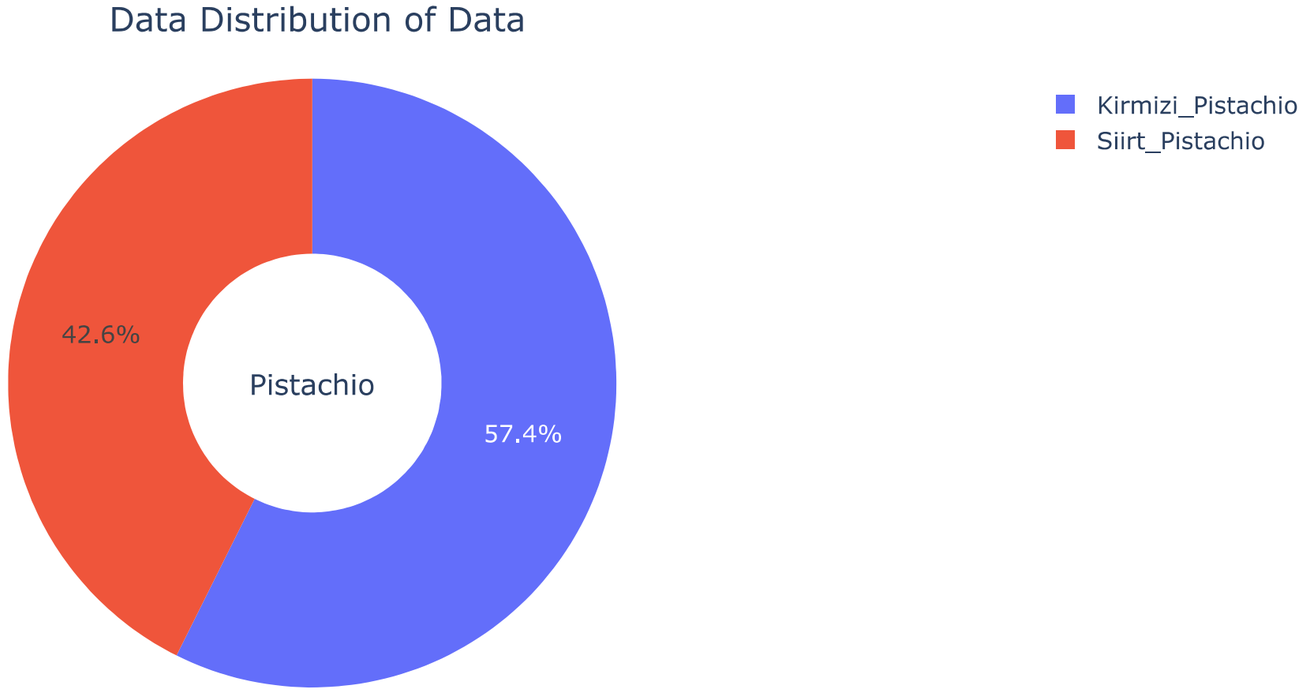
"Siirt_Pistachio" class contains 42.6% from data.

Clearly the data is not equaly distributed..

In [8]:
```
fig = go.Figure(go.Bar(
            x=  class_id_distributionTotal.values,
            y=class_id_distributionTotal.index,
            orientation='h'))

fig.update_layout(title='Data Distribution in Bars',font_size=15,title_x=0.45)


fig.show()
```



In [9]:
```
fig=px.pie(class_id_distributionTotal.head(10),values= 'ClassId', names=total_df['ClassId'].unique(),hole=0.425)
fig.update_layout(title='Data Distribution of Data',font_size=15,title_x=0.45,annotations=[dict(text='Pistachio',font_size=18, showarrow=False,height=800,width=700)])
fig.update_traces(textfont_size=15,textinfo='percent')
fig.show()
```



Splitting data into Train, Test, Val data set

using split folder so we can divide the data into three parts.

Training Set contain 80% of data

Test Set contain 10% of data

Validation Set contain 10% of data

In [10]:
```
splitfolders.ratio(data_dir, output="output", seed=101, ratio=(.8, .1, .1))
```

```
Copying files: 2148 files [00:19, 109.53 files/s]
```

Setting Path of the subfolder

In [11]:
```
train_path='./output/train/'
val_path='./output/val'
test_path='./output/test'
class_names=os.listdir(train_path)
class_names_val=os.listdir(val_path)
class_names_test=os.listdir(test_path)
```

In [12]:
```
train_image1 = glob.glob('./output/train/*/*.jpg')
Total_TrainImages = train_image1
print("Total number of training images: ", len(Total_TrainImages))


test_image1 = glob.glob('./output/test/*/*.jpg')
Total_TestImages = test_image1
print("Total number of test images: ", len(Total_TestImages))


Val_image1 = glob.glob('./output/val/*/*.jpg')

Total_ValImages = Val_image1
print("Total number of val images: ", len(Total_ValImages))
```

```
Total number of training images:  1717
Total number of test images:  217
Total number of val images:  214
```

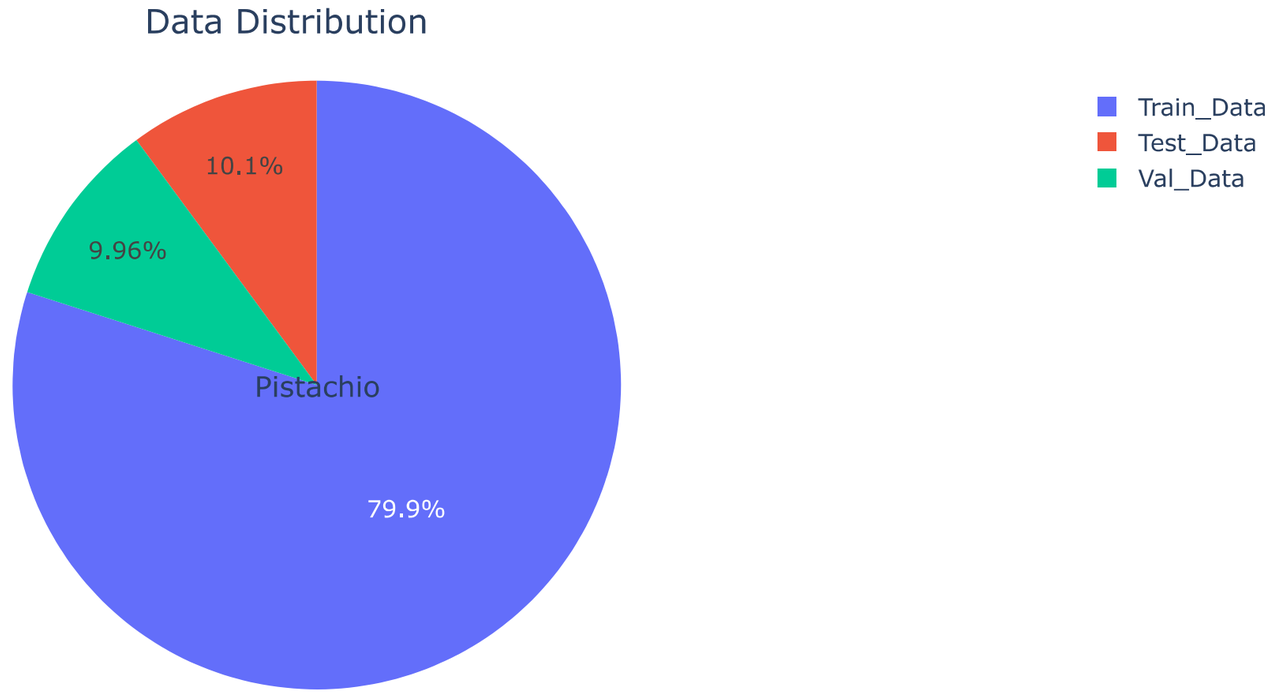
Below chart shows the perentage of data of three splitted data set..

"Trainng Set" contains 79.9% from data.

"Test Set" contains 10.1% from data.

"Val Set" contains 10% from data.

In [13]:
```
random_x = [len(Total_TrainImages), len(Total_TestImages), len(Total_ValImages)]
names = ['Train_Data', 'Test_Data', 'Val_Data']
fig = px.pie(values=random_x, names=names)
fig.update_layout(title='Data Distribution',font_size=15,title_x=0.45,annotations=[dict(text='Pistachio',font_size=18, showarrow=False,height=800,width=700)])
fig.update_traces(textfont_size=15,textinfo='percent')
fig.show()
```



In [14]:
```
train_image_names = pd.Series(Total_TrainImages)
train_df = pd.DataFrame()

# generate Filename field
train_df['Filename'] = train_image_names.map( lambda img_name: img_name.split("/")[-1])


# generate ClassId field
train_df['ClassId'] = train_image_names.map(lambda img_name: img_name.split("/")[-2])

train_df.head()
```

Out[14]:

	Filename	ClassId
0	siirt 544.jpg	Siirt_Pistachio
1	siirt 746.jpg	Siirt_Pistachio
2	siirt 298.jpg	Siirt_Pistachio
3	siirt 455.jpg	Siirt_Pistachio
4	siirt 420.jpg	Siirt_Pistachio

In [15]:
```
class_id_distribution_Train = train_df['ClassId'].value_counts()
class_id_distribution_Train.head(10)
```

Out[15]:
```
Kirmizi_Pistachio    985
Siirt_Pistachio      732
Name: ClassId, dtype: int64
```

Distribuation of Train Data

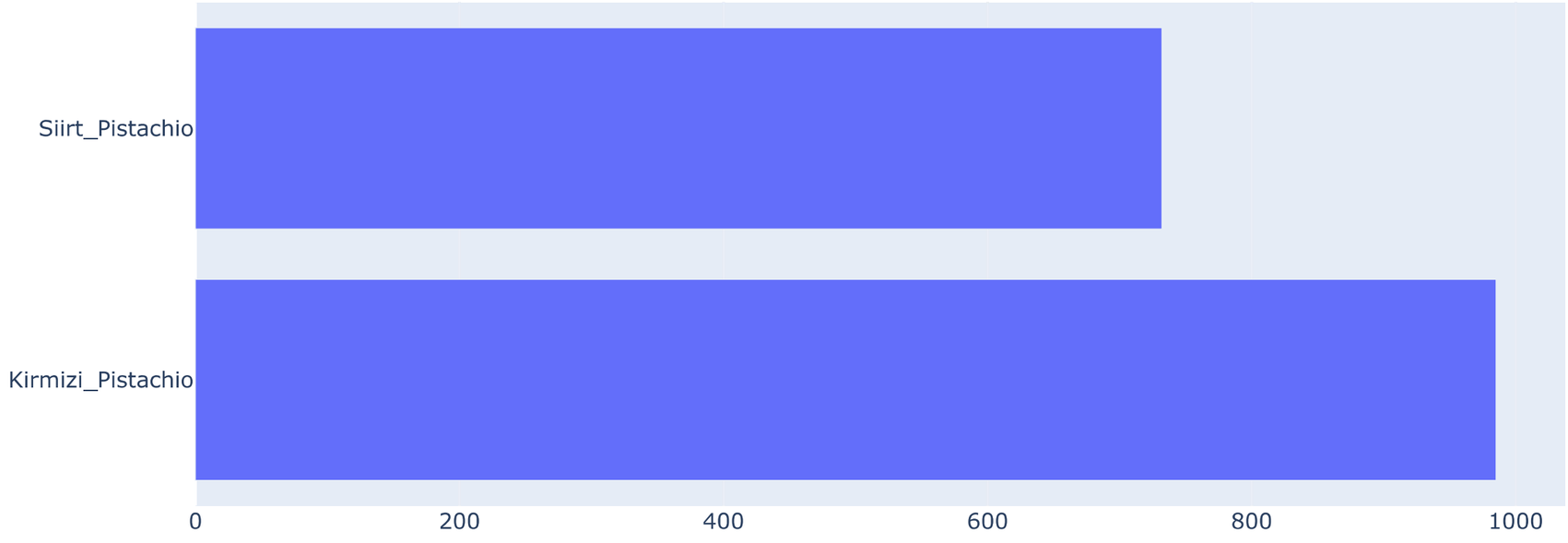
Below there are Two Charts of Training Data Set

Bar Chart.

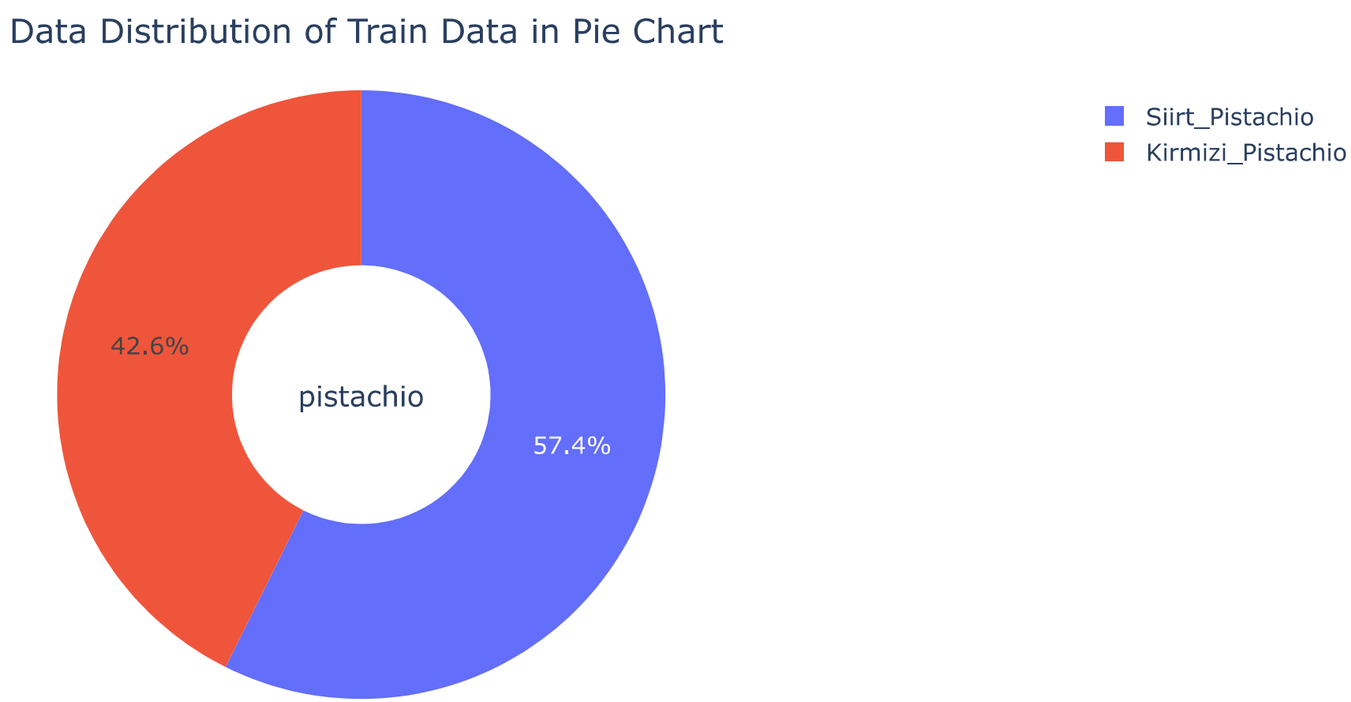
Pie Chart.

The data is not equally dsitributed..

In [16]:
```
fig = go.Figure(go.Bar(
            x= class_id_distribution_Train.values,
            y=class_id_distribution_Train.index,
            orientation='h'))

fig.update_layout(title='Data Distribution Of Train Data in Bars',font_size=15,title_x=0.45)

fig.show()
```



In [17]:
```
fig=px.pie(class_id_distribution_Train.head(10),values= 'ClassId', names=train_df['ClassId'].unique(),hole=0.425)
fig.update_layout(title='Data Distribution of Train Data in Pie Chart',font_size=15,title_x=0.45,annotations=[dict(text='pistachio',font_size=18, showarrow=False,height=800,width=700)])
fig.update_traces(textfont_size=15,textinfo='percent')
fig.show()
```



Distribution of Test Data

Below there are Two Charts of Test Data Set

Bar Chart

Pie Chart

The data is not equally dsitributed.

In [18]:
```
test_image_names = pd.Series(Total_TestImages)
test_df = pd.DataFrame()

# generate Filename field
test_df['Filename'] = test_image_names.map( lambda img_name: img_name.split("/")[-1])


# generate ClassId field
```

```
test_df['ClassId'] = test_image_names.map(lambda img_name: img_name.split("/")[-2])

test_df.head()
```

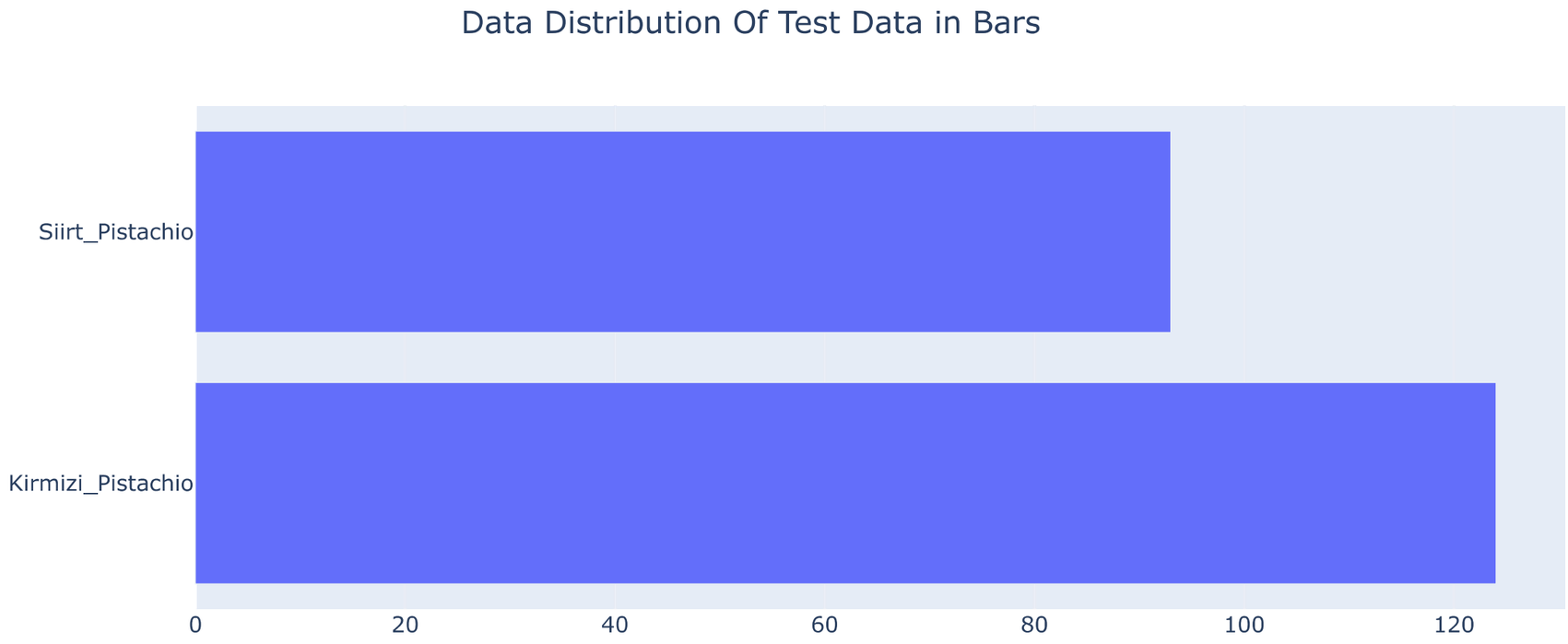
Out[18]:

	Filename	ClassId
0	siirt 265.jpg	Siirt_Pistachio
1	siirt 385.jpg	Siirt_Pistachio
2	siirt 443.jpg	Siirt_Pistachio
3	siirt 193.jpg	Siirt_Pistachio
4	siirt 296.jpg	Siirt_Pistachio

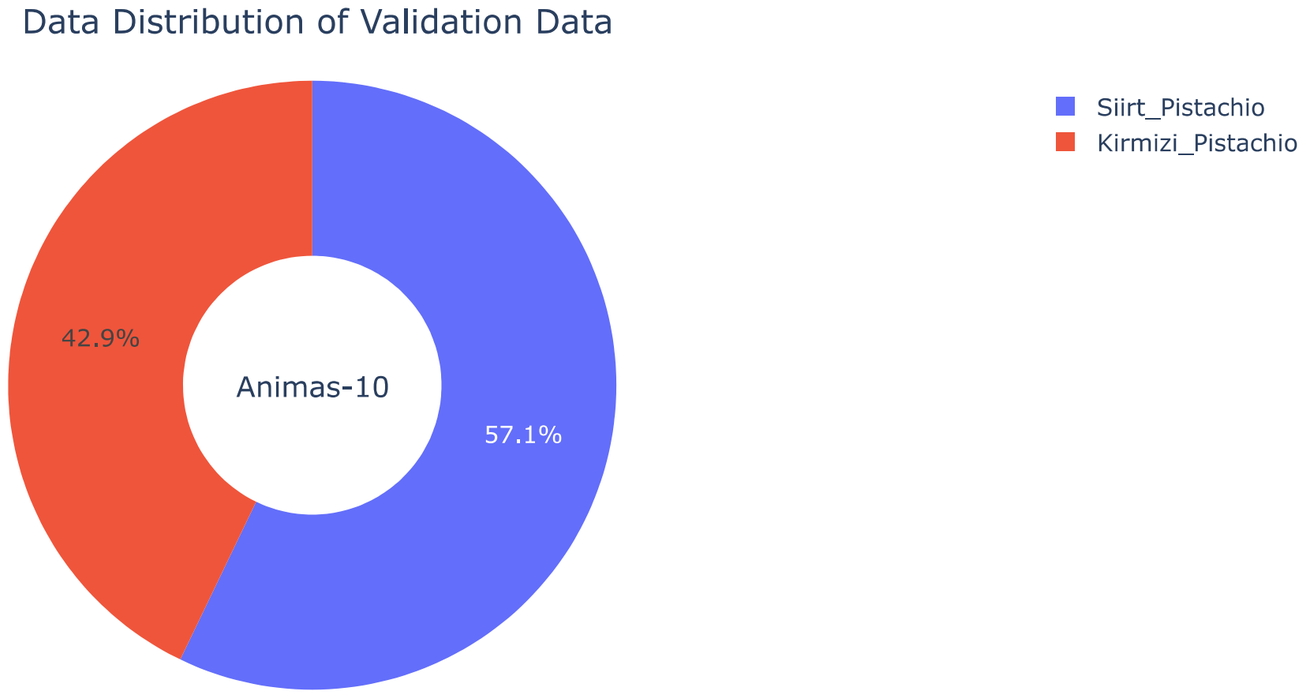
In [19]:
```
class_id_distribution_test = test_df['ClassId'].value_counts()
class_id_distribution_test.head(10)
```

Out[19]:
```
Kirmizi_Pistachio    124
Siirt_Pistachio       93
Name: ClassId, dtype: int64
```

In [20]:
```
fig = go.Figure(go.Bar(
            x=class_id_distribution_test.values,
            y=class_id_distribution_test.index,
            orientation='h'))

fig.update_layout(title='Data Distribution Of Test Data in Bars',font_size=15,title_x=0.45)

fig.show()
```



In [21]:
```
fig=px.pie(class_id_distribution_test.head(10),values= 'ClassId', names=test_df['ClassId'].unique(),hole=0.425)
fig.update_layout(title='Data Distribution of Validation Data',font_size=15,title_x=0.45,annotations=[dict(text='Animas-10',font_size=18, showarrow=False,height=800,width=700)])
fig.update_traces(textfont_size=15,textinfo='percent')
fig.show()
```



Dislaying Images

In [22]:
```
plot_df = train_df.sample(12).reset_index()
plt.figure(figsize=(15, 15))

for i in range(12):
    img_name = plot_df.loc[i, 'Filename']
    label_str = (plot_df.loc[i, 'ClassId'])
    plt.subplot(4,4,i+1)
    plt.imshow(plt.imread(os.path.join(train_path,label_str, img_name)))
    plt.title(label_str)
    plt.xticks([])
    plt.yticks([])
    plt.yticks([])
```

Image Data Generator

In [23]:
```
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(zoom_range=0.15,width_shift_range=0.2,height_shift_range=0.2,shear_range=0.15)
test_datagen = ImageDataGenerator()
val_datagen = ImageDataGenerator()
train_generator = train_datagen.flow_from_directory(train_path,target_size=(224, 224),batch_size=32,shuffle=True,class_mode='binary')
test_generator = test_datagen.flow_from_directory(test_path,target_size=(224,224),batch_size=32,shuffle=False,class_mode='binary')

val_generator = val_datagen.flow_from_directory(val_path,target_size=(224,224),batch_size=32,shuffle=False,class_mode='binary')
```

```
Found 1717 images belonging to 2 classes.
Found 217 images belonging to 2 classes.
Found 214 images belonging to 2 classes.
```

Efficent Net

Rethinking Model Scaling for Convolutional Neural Networks

In this paper, the authors systematically study model scaling and identify that carefully balancing network depth, width, and resolution can lead to better performance.

The most common way to scale up the network was by increasing the depth of the network. For example, ResNet-18 was scaled up to ResNet-150, but with only one dimension.

They do not develop a new network but instead develop the intuition that in order to achieve higher accuracy, the network should be scaled uniformly in all three dimensions.

They use the MobileNet as a baseline model and then develop EfficientNet B to B7 by using Neural Network Architecture Search.

In [24]:
```
from tensorflow.keras.applications import EfficientNetB0

model = EfficientNetB0(
      input_shape = (224,224,3),
      include_top = False,
      weights = 'imagenet'
    )
```

```
Downloading data from https://storage.googleapis.com/keras-applications/efficientnetb0_notop.h5
16711680/16705208 [==============================] - 0s 0us/step
16719872/16705208 [==============================] - 0s 0us/step
```

Fine Tuning

Fine-tuning not only involves replacing and retraining the classifier on top of the ConvNet with the new dataset but also fine-tuning the weights of the pretrained network by continuing the backpropagation.

It is possible to fine-tune all layers, or you can choose to freeze the early layers of the ConvNet.

The concept is that the early layers detect edges, colors, or what you can call early features, while the later layers capture specific details of the images.

In [25]:
```
model.trainable = True
```

In [26]:
```
for layer in model.layers[:-15]:
    layer.trainable=False
```

In [27]:
```
def print_layer_trainable():
    for layer in model.layers:
        print("{0}:\t{1}".format(layer.trainable, layer.name))
```

In [28]:
```
print_layer_trainable()
```

```
False:  input_1
False:  rescaling
False:  normalization
False:  stem_conv_pad
False:  stem_conv
False:  stem_bn
False:  stem_activation
False:  block1a_dwconv
False:  block1a_bn
False:  block1a_activation
False:  block1a_se_squeeze
False:  block1a_se_reshape
False:  block1a_se_reduce
False:  block1a_se_expand
False:  block1a_se_excite
False:  block1a_project_conv
False:  block1a_project_bn
False:  block2a_expand_conv
False:  block2a_expand_bn
False:  block2a_expand_activation
False:  block2a_dwconv_pad
False:  block2a_dwconv
False:  block2a_bn
False:  block2a_activation
False:  block2a_se_squeeze
False:  block2a_se_reshape
False:  block2a_se_reduce
False:  block2a_se_expand
False:  block2a_se_excite
False:  block2a_project_conv
False:  block2a_project_bn
False:  block2b_expand_conv
False:  block2b_expand_bn
False:  block2b_expand_activation
False:  block2b_dwconv
False:  block2b_bn
False:  block2b_activation
False:  block2b_se_squeeze
False:  block2b_se_reshape
False:  block2b_se_reduce
False:  block2b_se_expand
False:  block2b_se_excite
False:  block2b_project_conv
False:  block2b_project_bn
False:  block2b_drop
False:  block2b_add
False:  block3a_expand_conv
False:  block3a_expand_bn
False:  block3a_expand_activation
False:  block3a_dwconv_pad
False:  block3a_dwconv
False:  block3a_bn
False:  block3a_activation
False:  block3a_se_squeeze
False:  block3a_se_reshape
False:  block3a_se_reduce
False:  block3a_se_expand
False:  block3a_se_excite
False:  block3a_project_conv
False:  block3a_project_bn
```

```
False:  block3b_expand_conv
False:  block3b_expand_bn
False:  block3b_expand_activation
False:  block3b_dwconv
False:  block3b_bn
False:  block3b_activation
False:  block3b_se_squeeze
False:  block3b_se_reshape
False:  block3b_se_reduce
False:  block3b_se_expand
False:  block3b_se_excite
False:  block3b_project_conv
False:  block3b_project_bn
False:  block3b_drop
False:  block3b_add
False:  block4a_expand_conv
False:  block4a_expand_bn
False:  block4a_expand_activation
False:  block4a_dwconv_pad
False:  block4a_dwconv
False:  block4a_bn
False:  block4a_activation
False:  block4a_se_squeeze
False:  block4a_se_reshape
False:  block4a_se_reduce
False:  block4a_se_expand
False:  block4a_se_excite
False:  block4a_project_conv
False:  block4a_project_bn
False:  block4b_expand_conv
False:  block4b_expand_bn
False:  block4b_expand_activation
False:  block4b_dwconv
False:  block4b_bn
False:  block4b_activation
False:  block4b_se_squeeze
False:  block4b_se_reshape
False:  block4b_se_reduce
False:  block4b_se_expand
False:  block4b_se_excite
False:  block4b_project_conv
False:  block4b_project_bn
False:  block4b_drop
False:  block4b_add
False:  block4c_expand_conv
False:  block4c_expand_bn
False:  block4c_expand_activation
False:  block4c_dwconv
False:  block4c_bn
False:  block4c_activation
False:  block4c_se_squeeze
False:  block4c_se_reshape
False:  block4c_se_reduce
False:  block4c_se_expand
False:  block4c_se_excite
False:  block4c_project_conv
False:  block4c_project_bn
False:  block4c_drop
False:  block4c_add
False:  block5a_expand_conv
```

```
False:  block5a_expand_bn
False:  block5a_expand_activation
False:  block5a_dwconv
False:  block5a_bn
False:  block5a_activation
False:  block5a_se_squeeze
False:  block5a_se_reshape
False:  block5a_se_reduce
False:  block5a_se_expand
False:  block5a_se_excite
False:  block5a_project_conv
False:  block5a_project_bn
False:  block5b_expand_conv
False:  block5b_expand_bn
False:  block5b_expand_activation
False:  block5b_dwconv
False:  block5b_bn
False:  block5b_activation
False:  block5b_se_squeeze
False:  block5b_se_reshape
False:  block5b_se_reduce
False:  block5b_se_expand
False:  block5b_se_excite
False:  block5b_project_conv
False:  block5b_project_bn
False:  block5b_drop
False:  block5b_add
False:  block5c_expand_conv
False:  block5c_expand_bn
False:  block5c_expand_activation
False:  block5c_dwconv
False:  block5c_bn
False:  block5c_activation
False:  block5c_se_squeeze
False:  block5c_se_reshape
False:  block5c_se_reduce
False:  block5c_se_expand
False:  block5c_se_excite
False:  block5c_project_conv
False:  block5c_project_bn
False:  block5c_drop
False:  block5c_add
False:  block6a_expand_conv
False:  block6a_expand_bn
False:  block6a_expand_activation
False:  block6a_dwconv_pad
False:  block6a_dwconv
False:  block6a_bn
False:  block6a_activation
False:  block6a_se_squeeze
False:  block6a_se_reshape
False:  block6a_se_reduce
False:  block6a_se_expand
False:  block6a_se_excite
False:  block6a_project_conv
False:  block6a_project_bn
False:  block6b_expand_conv
False:  block6b_expand_bn
False:  block6b_expand_activation
False:  block6b_dwconv
```

```
False:  block6b_bn
False:  block6b_activation
False:  block6b_se_squeeze
False:  block6b_se_reshape
False:  block6b_se_reduce
False:  block6b_se_expand
False:  block6b_se_excite
False:  block6b_project_conv
False:  block6b_project_bn
False:  block6b_drop
False:  block6b_add
False:  block6c_expand_conv
False:  block6c_expand_bn
False:  block6c_expand_activation
False:  block6c_dwconv
False:  block6c_bn
False:  block6c_activation
False:  block6c_se_squeeze
False:  block6c_se_reshape
False:  block6c_se_reduce
False:  block6c_se_expand
False:  block6c_se_excite
False:  block6c_project_conv
False:  block6c_project_bn
False:  block6c_drop
False:  block6c_add
False:  block6d_expand_conv
False:  block6d_expand_bn
False:  block6d_expand_activation
False:  block6d_dwconv
False:  block6d_bn
False:  block6d_activation
False:  block6d_se_squeeze
False:  block6d_se_reshape
False:  block6d_se_reduce
False:  block6d_se_expand
False:  block6d_se_excite
False:  block6d_project_conv
False:  block6d_project_bn
False:  block6d_drop
False:  block6d_add
False:  block7a_expand_conv
True:   block7a_expand_bn
True:   block7a_expand_activation
True:   block7a_dwconv
True:   block7a_bn
True:   block7a_activation
True:   block7a_se_squeeze
True:   block7a_se_reshape
True:   block7a_se_reduce
True:   block7a_se_expand
True:   block7a_se_excite
True:   block7a_project_conv
True:   block7a_project_bn
True:   top_conv
True:   top_bn
True:   top_activation
```

In [29]:

```
from keras.layers import  Dropout
x = Flatten()(model.output)
```

```
x = Dropout(0.5)(x)
x = Dense(1, activation = "sigmoid")(x)

model = keras.Model(model.input, x)
model.compile(loss = "binary_crossentropy", optimizer = "adam", metrics = "accuracy")
```

In [30]:
```
print_layer_trainable()
```

```
False:  input_1
False:  rescaling
False:  normalization
False:  stem_conv_pad
False:  stem_conv
False:  stem_bn
False:  stem_activation
False:  block1a_dwconv
False:  block1a_bn
False:  block1a_activation
False:  block1a_se_squeeze
False:  block1a_se_reshape
False:  block1a_se_reduce
False:  block1a_se_expand
False:  block1a_se_excite
False:  block1a_project_conv
False:  block1a_project_bn
False:  block2a_expand_conv
False:  block2a_expand_bn
False:  block2a_expand_activation
False:  block2a_dwconv_pad
False:  block2a_dwconv
False:  block2a_bn
False:  block2a_activation
False:  block2a_se_squeeze
False:  block2a_se_reshape
False:  block2a_se_reduce
False:  block2a_se_expand
False:  block2a_se_excite
False:  block2a_project_conv
False:  block2a_project_bn
False:  block2b_expand_conv
False:  block2b_expand_bn
False:  block2b_expand_activation
False:  block2b_dwconv
False:  block2b_bn
False:  block2b_activation
False:  block2b_se_squeeze
False:  block2b_se_reshape
False:  block2b_se_reduce
False:  block2b_se_expand
False:  block2b_se_excite
False:  block2b_project_conv
False:  block2b_project_bn
False:  block2b_drop
False:  block2b_add
False:  block3a_expand_conv
False:  block3a_expand_bn
False:  block3a_expand_activation
False:  block3a_dwconv_pad
False:  block3a_dwconv
False:  block3a_bn
False:  block3a_activation
False:  block3a_se_squeeze
False:  block3a_se_reshape
False:  block3a_se_reduce
False:  block3a_se_expand
False:  block3a_se_excite
False:  block3a_project_conv
False:  block3a_project_bn
```

```
False:  block3b_expand_conv
False:  block3b_expand_bn
False:  block3b_expand_activation
False:  block3b_dwconv
False:  block3b_bn
False:  block3b_activation
False:  block3b_se_squeeze
False:  block3b_se_reshape
False:  block3b_se_reduce
False:  block3b_se_expand
False:  block3b_se_excite
False:  block3b_project_conv
False:  block3b_project_bn
False:  block3b_drop
False:  block3b_add
False:  block4a_expand_conv
False:  block4a_expand_bn
False:  block4a_expand_activation
False:  block4a_dwconv_pad
False:  block4a_dwconv
False:  block4a_bn
False:  block4a_activation
False:  block4a_se_squeeze
False:  block4a_se_reshape
False:  block4a_se_reduce
False:  block4a_se_expand
False:  block4a_se_excite
False:  block4a_project_conv
False:  block4a_project_bn
False:  block4b_expand_conv
False:  block4b_expand_bn
False:  block4b_expand_activation
False:  block4b_dwconv
False:  block4b_bn
False:  block4b_activation
False:  block4b_se_squeeze
False:  block4b_se_reshape
False:  block4b_se_reduce
False:  block4b_se_expand
False:  block4b_se_excite
False:  block4b_project_conv
False:  block4b_project_bn
False:  block4b_drop
False:  block4b_add
False:  block4c_expand_conv
False:  block4c_expand_bn
False:  block4c_expand_activation
False:  block4c_dwconv
False:  block4c_bn
False:  block4c_activation
False:  block4c_se_squeeze
False:  block4c_se_reshape
False:  block4c_se_reduce
False:  block4c_se_expand
False:  block4c_se_excite
False:  block4c_project_conv
False:  block4c_project_bn
False:  block4c_drop
False:  block4c_add
False:  block5a_expand_conv
```

```
False:  block5a_expand_bn
False:  block5a_expand_activation
False:  block5a_dwconv
False:  block5a_bn
False:  block5a_activation
False:  block5a_se_squeeze
False:  block5a_se_reshape
False:  block5a_se_reduce
False:  block5a_se_expand
False:  block5a_se_excite
False:  block5a_project_conv
False:  block5a_project_bn
False:  block5b_expand_conv
False:  block5b_expand_bn
False:  block5b_expand_activation
False:  block5b_dwconv
False:  block5b_bn
False:  block5b_activation
False:  block5b_se_squeeze
False:  block5b_se_reshape
False:  block5b_se_reduce
False:  block5b_se_expand
False:  block5b_se_excite
False:  block5b_project_conv
False:  block5b_project_bn
False:  block5b_drop
False:  block5b_add
False:  block5c_expand_conv
False:  block5c_expand_bn
False:  block5c_expand_activation
False:  block5c_dwconv
False:  block5c_bn
False:  block5c_activation
False:  block5c_se_squeeze
False:  block5c_se_reshape
False:  block5c_se_reduce
False:  block5c_se_expand
False:  block5c_se_excite
False:  block5c_project_conv
False:  block5c_project_bn
False:  block5c_drop
False:  block5c_add
False:  block6a_expand_conv
False:  block6a_expand_bn
False:  block6a_expand_activation
False:  block6a_dwconv_pad
False:  block6a_dwconv
False:  block6a_bn
False:  block6a_activation
False:  block6a_se_squeeze
False:  block6a_se_reshape
False:  block6a_se_reduce
False:  block6a_se_expand
False:  block6a_se_excite
False:  block6a_project_conv
False:  block6a_project_bn
False:  block6b_expand_conv
False:  block6b_expand_bn
False:  block6b_expand_activation
False:  block6b_dwconv
```

```
False:  block6b_bn
False:  block6b_activation
False:  block6b_se_squeeze
False:  block6b_se_reshape
False:  block6b_se_reduce
False:  block6b_se_expand
False:  block6b_se_excite
False:  block6b_project_conv
False:  block6b_project_bn
False:  block6b_drop
False:  block6b_add
False:  block6c_expand_conv
False:  block6c_expand_bn
False:  block6c_expand_activation
False:  block6c_dwconv
False:  block6c_bn
False:  block6c_activation
False:  block6c_se_squeeze
False:  block6c_se_reshape
False:  block6c_se_reduce
False:  block6c_se_expand
False:  block6c_se_excite
False:  block6c_project_conv
False:  block6c_project_bn
False:  block6c_drop
False:  block6c_add
False:  block6d_expand_conv
False:  block6d_expand_bn
False:  block6d_expand_activation
False:  block6d_dwconv
False:  block6d_bn
False:  block6d_activation
False:  block6d_se_squeeze
False:  block6d_se_reshape
False:  block6d_se_reduce
False:  block6d_se_expand
False:  block6d_se_excite
False:  block6d_project_conv
False:  block6d_project_bn
False:  block6d_drop
False:  block6d_add
False:  block7a_expand_conv
True:   block7a_expand_bn
True:   block7a_expand_activation
True:   block7a_dwconv
True:   block7a_bn
True:   block7a_activation
True:   block7a_se_squeeze
True:   block7a_se_reshape
True:   block7a_se_reduce
True:   block7a_se_expand
True:   block7a_se_excite
True:   block7a_project_conv
True:   block7a_project_bn
True:   top_conv
True:   top_bn
True:   top_activation
True:   flatten
True:   dropout
True:   dense
```

Visual Respresntation of Model

In [ ]:
```
from tensorflow.keras.utils import plot_model
from IPython.display import Image
plot_model(model, to_file='convnet.png', show_shapes=True,show_layer_names=True)
Image(filename='convnet.png')
```

In [32]:
```
!pip install visualkeras
import visualkeras
visualkeras.layered_view(model, legend=True)
```

```
Collecting visualkeras
  Downloading visualkeras-0.0.2-py3-none-any.whl (12 kB)
Collecting aggdraw>=1.3.11
  Downloading aggdraw-1.3.16-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (989 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 989.8/989.8 kB 17.8 MB/s eta 0:00:00a 0:00:01
Requirement already satisfied: numpy>=1.18.1 in /opt/conda/lib/python3.7/site-packages (from visualkeras) (1.21.6)
Requirement already satisfied: pillow>=6.2.0 in /opt/conda/lib/python3.7/site-packages (from visualkeras) (9.1.0)
Installing collected packages: aggdraw, visualkeras
Successfully installed aggdraw-1.3.16 visualkeras-0.0.2
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
```

Out[32]:

Call Backs

A callback is an object capable of executing actions at different stages of training, such as at the beginning or end of an epoch, before or after a single batch, and more.

In [33]:
```
es=EarlyStopping(monitor='val_accuracy', mode='max', verbose=1, patience=20)
```

In [34]:
```
mc = ModelCheckpoint('model.h5', monitor='val_accuracy', mode='max',save_best_only=True )
```

Train The model

In [35]:
```
H = model.fit_generator(train_generator,validation_data=val_generator,epochs=50,verbose=1,callbacks=[mc,es])
```

```
/opt/conda/lib/python3.7/site-packages/keras/engine/training.py:1972: UserWarning:

`Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.

Epoch 1/50
54/54 [==============================] - 37s 467ms/step - loss: 0.5472 - accuracy: 0.8451 - val_loss: 4.3422 - val_accuracy: 0.5000
/opt/conda/lib/python3.7/site-packages/keras/utils/generic_utils.py:497: CustomMaskWarning:

Custom mask layers require a config and must override get_config. When loading, the custom mask layer must be passed to the custom_objects argument.
```

```
Epoch 2/50
54/54 [==============================] - 24s 442ms/step - loss: 0.3197 - accuracy: 0.9301 - val_loss: 0.2252 - val_accuracy: 0.9439
Epoch 3/50
54/54 [==============================] - 24s 444ms/step - loss: 0.2601 - accuracy: 0.9359 - val_loss: 0.3765 - val_accuracy: 0.9112
Epoch 4/50
54/54 [==============================] - 24s 446ms/step - loss: 0.2450 - accuracy: 0.9383 - val_loss: 1.0266 - val_accuracy: 0.8318
Epoch 5/50
54/54 [==============================] - 24s 446ms/step - loss: 0.2125 - accuracy: 0.9581 - val_loss: 0.3418 - val_accuracy: 0.9439
Epoch 6/50
54/54 [==============================] - 24s 442ms/step - loss: 0.1981 - accuracy: 0.9592 - val_loss: 0.2939 - val_accuracy: 0.9439
Epoch 7/50
54/54 [==============================] - 24s 443ms/step - loss: 0.1455 - accuracy: 0.9656 - val_loss: 0.3353 - val_accuracy: 0.9486
Epoch 8/50
54/54 [==============================] - 24s 446ms/step - loss: 0.1236 - accuracy: 0.9755 - val_loss: 0.4418 - val_accuracy: 0.9206
Epoch 9/50
54/54 [==============================] - 24s 444ms/step - loss: 0.1139 - accuracy: 0.9715 - val_loss: 0.7168 - val_accuracy: 0.9206
Epoch 10/50
54/54 [==============================] - 24s 446ms/step - loss: 0.1044 - accuracy: 0.9726 - val_loss: 1.7725 - val_accuracy: 0.7897
Epoch 11/50
54/54 [==============================] - 24s 447ms/step - loss: 0.1219 - accuracy: 0.9761 - val_loss: 1.5888 - val_accuracy: 0.8598
Epoch 12/50
54/54 [==============================] - 25s 454ms/step - loss: 0.1332 - accuracy: 0.9744 - val_loss: 1.3745 - val_accuracy: 0.8551
Epoch 13/50
54/54 [==============================] - 24s 440ms/step - loss: 0.1754 - accuracy: 0.9656 - val_loss: 1.1788 - val_accuracy: 0.8785
Epoch 14/50
54/54 [==============================] - 24s 445ms/step - loss: 0.2238 - accuracy: 0.9610 - val_loss: 0.7897 - val_accuracy: 0.8925
Epoch 15/50
54/54 [==============================] - 24s 447ms/step - loss: 0.1372 - accuracy: 0.9691 - val_loss: 0.6571 - val_accuracy: 0.9206
Epoch 16/50
54/54 [==============================] - 24s 445ms/step - loss: 0.1195 - accuracy: 0.9779 - val_loss: 0.3645 - val_accuracy: 0.9439
Epoch 17/50
54/54 [==============================] - 24s 447ms/step - loss: 0.1596 - accuracy: 0.9668 - val_loss: 0.4450 - val_accuracy: 0.9533
Epoch 18/50
54/54 [==============================] - 24s 446ms/step - loss: 0.1272 - accuracy: 0.9755 - val_loss: 0.4293 - val_accuracy: 0.9439
Epoch 19/50
54/54 [==============================] - 24s 449ms/step - loss: 0.1278 - accuracy: 0.9773 - val_loss: 1.0945 - val_accuracy: 0.8925
Epoch 20/50
54/54 [==============================] - 24s 449ms/step - loss: 0.1265 - accuracy: 0.9779 - val_loss: 1.1728 - val_accuracy: 0.8925
Epoch 21/50
54/54 [==============================] - 24s 450ms/step - loss: 0.1430 - accuracy: 0.9674 - val_loss: 0.9057 - val_accuracy: 0.9112
Epoch 22/50
54/54 [==============================] - 24s 443ms/step - loss: 0.1995 - accuracy: 0.9662 - val_loss: 0.6406 - val_accuracy: 0.9533
Epoch 23/50
54/54 [==============================] - 24s 447ms/step - loss: 0.1027 - accuracy: 0.9796 - val_loss: 0.7649 - val_accuracy: 0.9393
Epoch 24/50
54/54 [==============================] - 24s 443ms/step - loss: 0.1836 - accuracy: 0.9697 - val_loss: 1.1105 - val_accuracy: 0.8879
Epoch 25/50
54/54 [==============================] - 24s 445ms/step - loss: 0.1108 - accuracy: 0.9796 - val_loss: 0.5089 - val_accuracy: 0.9393
Epoch 26/50
54/54 [==============================] - 24s 443ms/step - loss: 0.0922 - accuracy: 0.9831 - val_loss: 0.7295 - val_accuracy: 0.9159
Epoch 27/50
54/54 [==============================] - 24s 441ms/step - loss: 0.0962 - accuracy: 0.9825 - val_loss: 0.7409 - val_accuracy: 0.9299
Epoch 28/50
54/54 [==============================] - 24s 440ms/step - loss: 0.1371 - accuracy: 0.9750 - val_loss: 1.6375 - val_accuracy: 0.8879
Epoch 29/50
54/54 [==============================] - 24s 441ms/step - loss: 0.0577 - accuracy: 0.9889 - val_loss: 1.8766 - val_accuracy: 0.8785
Epoch 30/50
54/54 [==============================] - 24s 440ms/step - loss: 0.0836 - accuracy: 0.9878 - val_loss: 0.4766 - val_accuracy: 0.9393
Epoch 31/50
54/54 [==============================] - 24s 441ms/step - loss: 0.1607 - accuracy: 0.9779 - val_loss: 0.9465 - val_accuracy: 0.8785
```

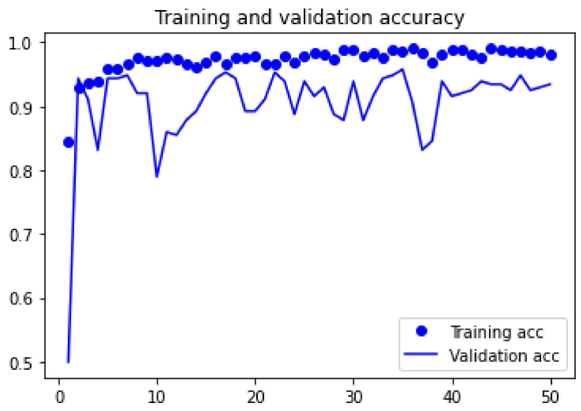
```
Epoch 32/50
54/54 [==============================] - 24s 441ms/step - loss: 0.1191 - accuracy: 0.9843 - val_loss: 0.6744 - val_accuracy: 0.9159
Epoch 33/50
54/54 [==============================] - 24s 440ms/step - loss: 0.1277 - accuracy: 0.9767 - val_loss: 0.3554 - val_accuracy: 0.9439
Epoch 34/50
54/54 [==============================] - 24s 443ms/step - loss: 0.0745 - accuracy: 0.9889 - val_loss: 0.2407 - val_accuracy: 0.9486
Epoch 35/50
54/54 [==============================] - 24s 440ms/step - loss: 0.0568 - accuracy: 0.9866 - val_loss: 0.2009 - val_accuracy: 0.9579
Epoch 36/50
54/54 [==============================] - 24s 439ms/step - loss: 0.0348 - accuracy: 0.9901 - val_loss: 0.7937 - val_accuracy: 0.9065
Epoch 37/50
54/54 [==============================] - 24s 443ms/step - loss: 0.0872 - accuracy: 0.9837 - val_loss: 2.0375 - val_accuracy: 0.8318
Epoch 38/50
54/54 [==============================] - 24s 443ms/step - loss: 0.1823 - accuracy: 0.9697 - val_loss: 1.7493 - val_accuracy: 0.8458
Epoch 39/50
54/54 [==============================] - 24s 442ms/step - loss: 0.1217 - accuracy: 0.9825 - val_loss: 0.4959 - val_accuracy: 0.9393
Epoch 40/50
54/54 [==============================] - 24s 443ms/step - loss: 0.0466 - accuracy: 0.9889 - val_loss: 0.6589 - val_accuracy: 0.9159
Epoch 41/50
54/54 [==============================] - 24s 442ms/step - loss: 0.0701 - accuracy: 0.9889 - val_loss: 0.5505 - val_accuracy: 0.9206
Epoch 42/50
54/54 [==============================] - 24s 445ms/step - loss: 0.1201 - accuracy: 0.9808 - val_loss: 0.5943 - val_accuracy: 0.9252
Epoch 43/50
54/54 [==============================] - 24s 441ms/step - loss: 0.1516 - accuracy: 0.9755 - val_loss: 0.6634 - val_accuracy: 0.9393
Epoch 44/50
54/54 [==============================] - 24s 440ms/step - loss: 0.0572 - accuracy: 0.9907 - val_loss: 0.8353 - val_accuracy: 0.9346
Epoch 45/50
54/54 [==============================] - 24s 440ms/step - loss: 0.0803 - accuracy: 0.9889 - val_loss: 0.5180 - val_accuracy: 0.9346
Epoch 46/50
54/54 [==============================] - 24s 443ms/step - loss: 0.0738 - accuracy: 0.9872 - val_loss: 0.8243 - val_accuracy: 0.9252
Epoch 47/50
54/54 [==============================] - 24s 440ms/step - loss: 0.0694 - accuracy: 0.9872 - val_loss: 0.5069 - val_accuracy: 0.9486
Epoch 48/50
54/54 [==============================] - 24s 443ms/step - loss: 0.0921 - accuracy: 0.9849 - val_loss: 0.5481 - val_accuracy: 0.9252
Epoch 49/50
54/54 [==============================] - 24s 442ms/step - loss: 0.0772 - accuracy: 0.9866 - val_loss: 0.5319 - val_accuracy: 0.9299
Epoch 50/50
54/54 [==============================] - 24s 438ms/step - loss: 0.1463 - accuracy: 0.9814 - val_loss: 0.6680 - val_accuracy: 0.9346
```

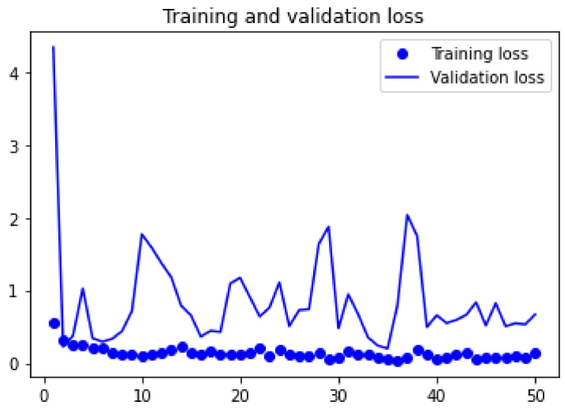
Plotting the Loss and Accuracy

In [36]:

```
acc = H.history['accuracy']
val_acc = H.history['val_accuracy']
loss = H.history['loss']
val_loss = H.history['val_loss']



epochs = range(1, len(acc) + 1)

plt.plot(epochs, acc, 'bo', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.legend()

plt.figure()

plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
```

```
plt.title('Training and validation loss')
plt.legend()

plt.show()
```





Model Evaluation

In [37]:
```
test_loss, test_acc = model.evaluate(test_generator, steps=len(test_generator), verbose=1)
print('Loss: %.3f' % (test_loss * 100.0))
print('Accuracy: %.3f' % (test_acc * 100.0))
```

```
7/7 [==============================] - 1s 159ms/step - loss: 0.9502 - accuracy: 0.9355
Loss: 95.018
Accuracy: 93.548
```

Classifiaction Report

In [38]:
```
from sklearn.metrics import classification_report
```

In [39]:
```
y_val = test_generator.classes
y_pred = model.predict(test_generator)
y_pred = np.argmax(y_pred,axis=1)
```

In [40]:
```
print(classification_report(y_val,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.57      1.00      0.73       124
           1       0.00      0.00      0.00        93

    accuracy                           0.57       217
   macro avg       0.29      0.50      0.36       217
weighted avg       0.33      0.57      0.42       217
```

```
/opt/conda/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning:

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.

/opt/conda/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning:

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.

/opt/conda/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning:

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
```

In [41]:
```
class_indices = test_generator.class_indices
indices = {v:k for k,v in class_indices.items()}
```

In [42]:
```
filenames = test_generator.filenames
```

In [43]:
```
val_df = pd.DataFrame()
val_df['filename'] = filenames
val_df['actual'] = y_val
val_df['predicted'] = y_pred
val_df['actual'] = val_df['actual'].apply(lambda x: indices[x])
val_df['predicted'] = val_df['predicted'].apply(lambda x: indices[x])
val_df.loc[val_df['actual']==val_df['predicted'],'Same'] = True
val_df.loc[val_df['actual']!=val_df['predicted'],'Same'] = False
val_df.head(10)
```

Out[43]:

	filename	actual	predicted	Same
0	Kirmizi_Pistachio/kirmizi (20).jpg	Kirmizi_Pistachio	Kirmizi_Pistachio	True
1	Kirmizi_Pistachio/kirmizi (21).jpg	Kirmizi_Pistachio	Kirmizi_Pistachio	True
2	Kirmizi_Pistachio/kirmizi (27).jpg	Kirmizi_Pistachio	Kirmizi_Pistachio	True
3	Kirmizi_Pistachio/kirmizi (31).jpg	Kirmizi_Pistachio	Kirmizi_Pistachio	True
4	Kirmizi_Pistachio/kirmizi (35).jpg	Kirmizi_Pistachio	Kirmizi_Pistachio	True
5	Kirmizi_Pistachio/kirmizi (5).jpg	Kirmizi_Pistachio	Kirmizi_Pistachio	True
6	Kirmizi_Pistachio/kirmizi 1.jpg	Kirmizi_Pistachio	Kirmizi_Pistachio	True
7	Kirmizi_Pistachio/kirmizi 1002.jpg	Kirmizi_Pistachio	Kirmizi_Pistachio	True
8	Kirmizi_Pistachio/kirmizi 1021.jpg	Kirmizi_Pistachio	Kirmizi_Pistachio	True
9	Kirmizi_Pistachio/kirmizi 1029.jpg	Kirmizi_Pistachio	Kirmizi_Pistachio	True

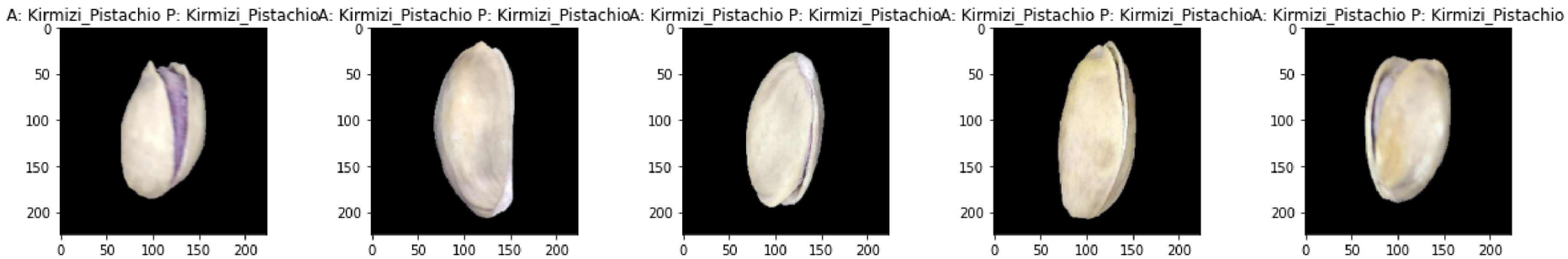
In [44]:
```
val_df = val_df.sample(frac=1).reset_index(drop=True)
```

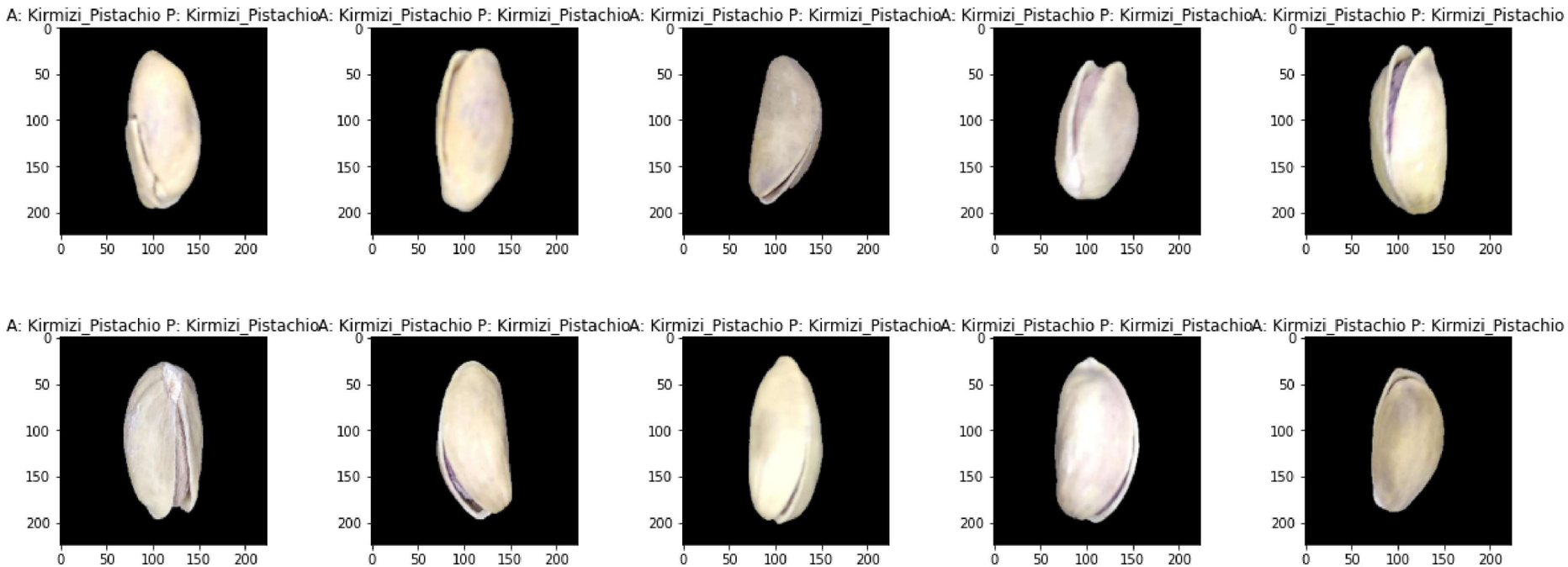
Prediction Comparison

In [45]:
```
from tensorflow.keras.preprocessing.image import ImageDataGenerator, array_to_img, img_to_array, load_img
img_size = 224
def readImage(path):
    img = load_img(path,color_mode='rgb',target_size=(img_size,img_size))
    img = img_to_array(img)
    img = img/255.

    return img

def display_images(temp_df):
    temp_df = temp_df.reset_index(drop=True)
    plt.figure(figsize = (20 , 20))
    n = 0
    for i in range(15):
        n+=1
        plt.subplot(5 , 5, n)
        plt.subplots_adjust(hspace = 0.5 , wspace = 0.3)
        image = readImage(f"../input/Pistachio_Image_Dataset/Pistachio_Image_Dataset/{temp_df.filename[i]}")
        plt.imshow(image)
        plt.title(f'A: {temp_df.actual[i]} P: {temp_df.predicted[i]}')
```

Correctly Classified

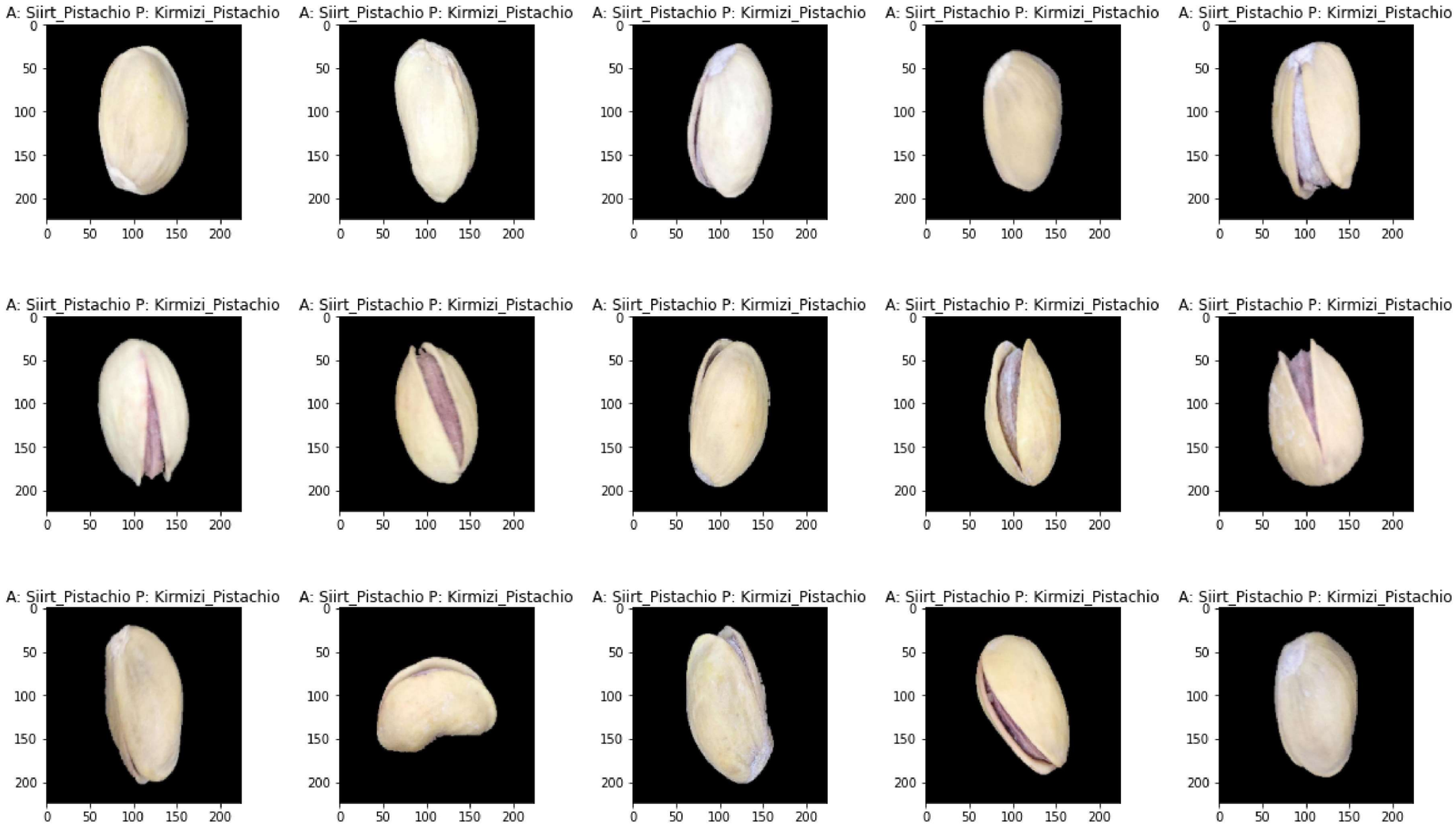
In [46]:
```
display_images(val_df[val_df['Same']==True])
```





Miss Classified

In [47]:
```
display_images(val_df[val_df['Same']!=True])
```



Confusion Matrix

In [48]:
```
cm = confusion_matrix(y_true=y_val, y_pred=y_pred)
```

In [49]:
```
def plot_confusion_matrix(cm, classes,normalize=False,title='Confusion matrix',cmap=plt.cm.Blues):

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
```

```
        print('Confusion matrix, without normalization')
        print(cm)

    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
        horizontalalignment="center",
        color="white" if cm[i, j] > thresh else "black")
        plt.tight_layout()
        plt.ylabel('True label')
        plt.xlabel('Predicted label')
```

In [50]:
```
cm_plot_labels = ['1_Kirmizi_Pistachio', '2_Siirt_Pistachio']

plot_confusion_matrix(cm=cm, classes=cm_plot_labels, title='Confusion Matrix')
```

```
Confusion matrix, without normalization
[[124   0]
 [ 93   0]]
```