FLOWER RECOGNITION

Rimple Poonia Dabas
Sweta Joshi
Sakshi Dalal

Data Collection:

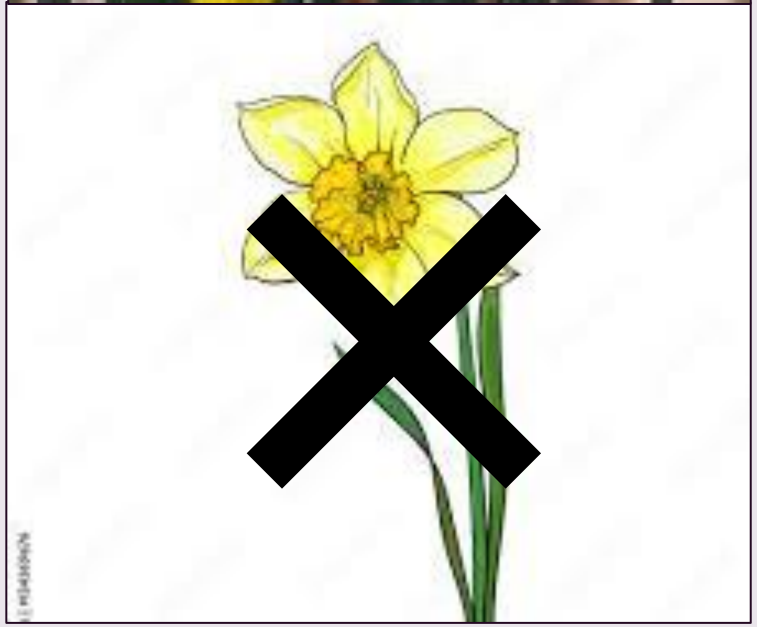
- Bougainvillea
- Daffodil
- Dahlia
- Foxglove
- Hibiscus
- Hydrangea
- Orchid
- Rose
- Sunflower
- Tulip

Google Images: Web scraping (Selenium and Chrome driver).

Flickr: API (good API call limit 3600 queries per hour), used flickrapi python package.

Pixabay: API (100 requests per minute limit)

Data Cleaning



Data preprocessing

- **Labeling**
- **Saving as .npy**
- **Resizing**
- **Rescaling**

	img_folder	img_label
0	../Combined flowers/bougainvillea	bougainvillea
1	../Combined flowers/daffodil	daffodil
2	../Combined flowers/dahlia	dahlia
3	../Combined flowers/foxglove	foxglove
4	../Combined flowers/hibiscus	hibiscus
5	../Combined flowers/hydrangea	hydrangea
6	../Combined flowers/orchid	orchid
7	../Combined flowers/rose	rose
8	../Combined flowers/sunflower	sunflower
9	../Combined flowers/tulip	tulip

```
def make_train_data(flower_type,DIR):
    for img in tqdm(os.listdir(DIR)):
        label = assign_label(img,flower_type)
        path = os.path.join(DIR,img)
        img = cv2.imread(path, cv2.IMREAD_COLOR)
        img = cv2.resize(img, (256, 256))

        X.append(np.array(img))
        Z.append(str(label))
```

```
X = np.load("dataset/data-256.npy")
labels = np.load("dataset/label-256.npy")
```

```
from sklearn.preprocessing import LabelEncoder
from keras.utils import to_categorical

labelEncoder = LabelEncoder()
y = labelEncoder.fit_transform(labels)
y = to_categorical(y, 10)
```

Model Architecture: Convolutional Neural Network

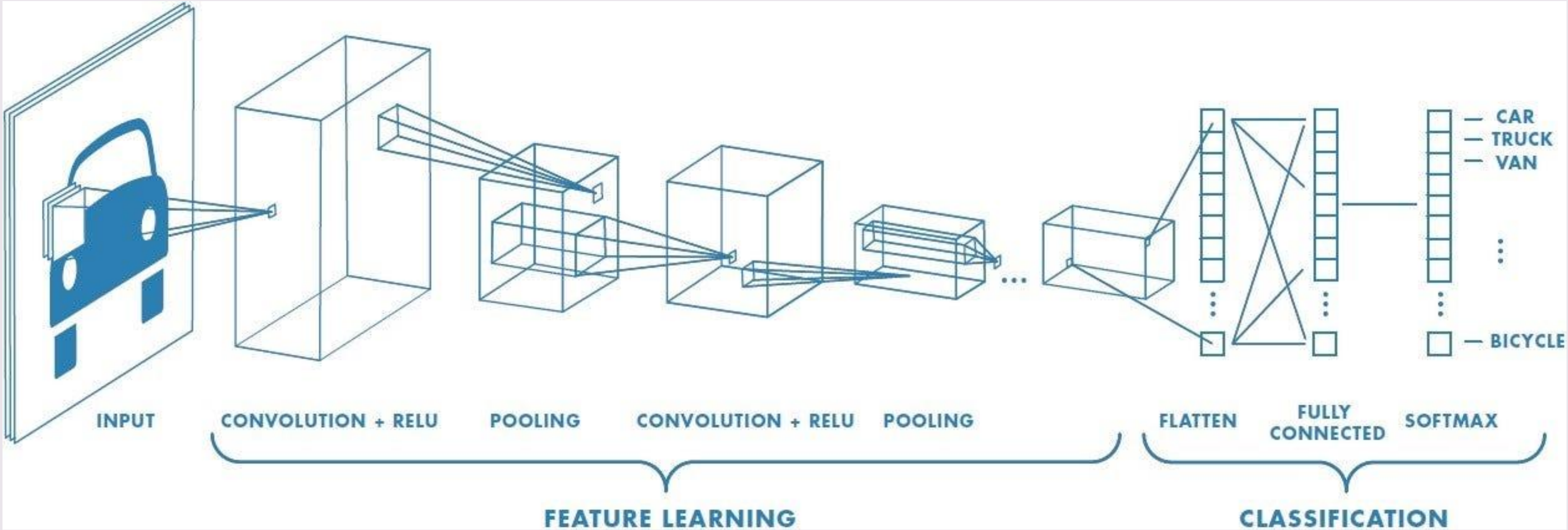
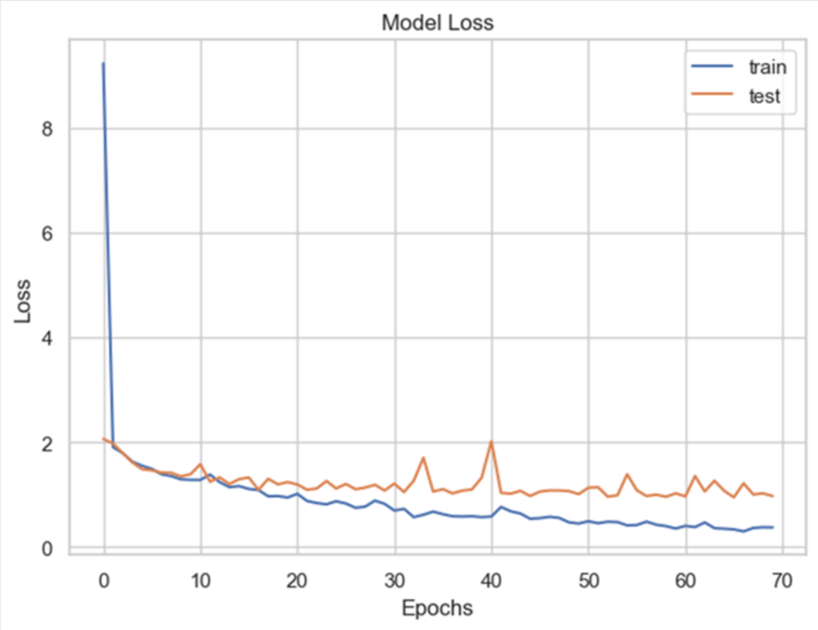


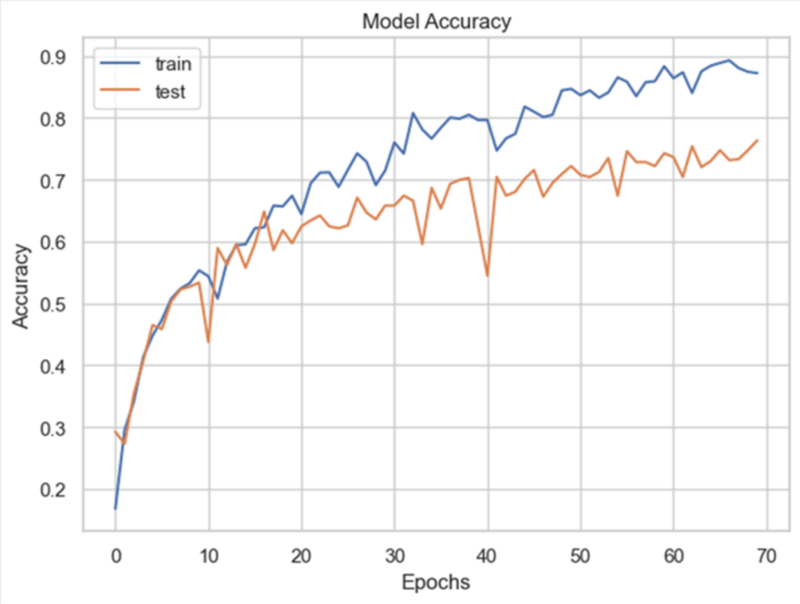
Image source : https://www.interviewbit.com/blog/cnn-architecture/

Model Comparison

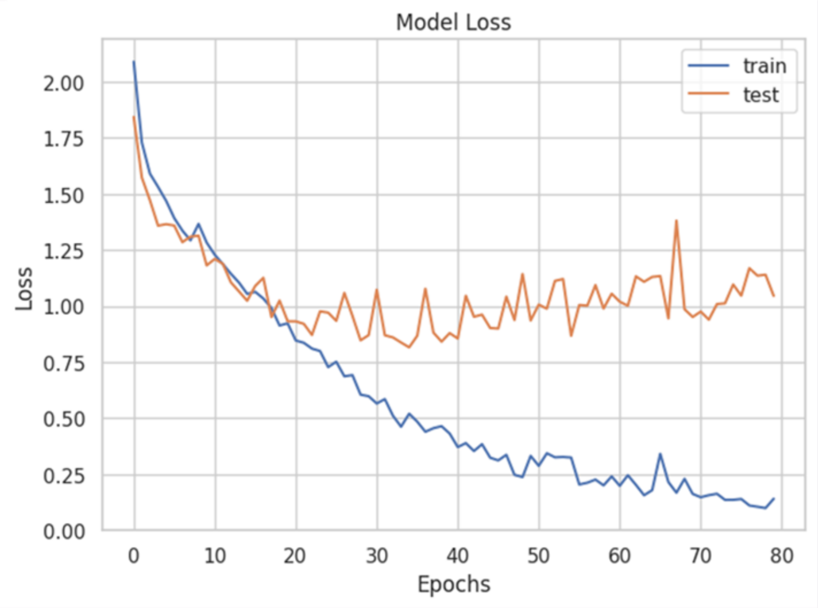
Batch Size	Epoch	Parameters	Accuracy
32	50	9,599,882	60
64	70	4,146,314	74
64	70	3,393,770	76
64	80	4,146,314	79

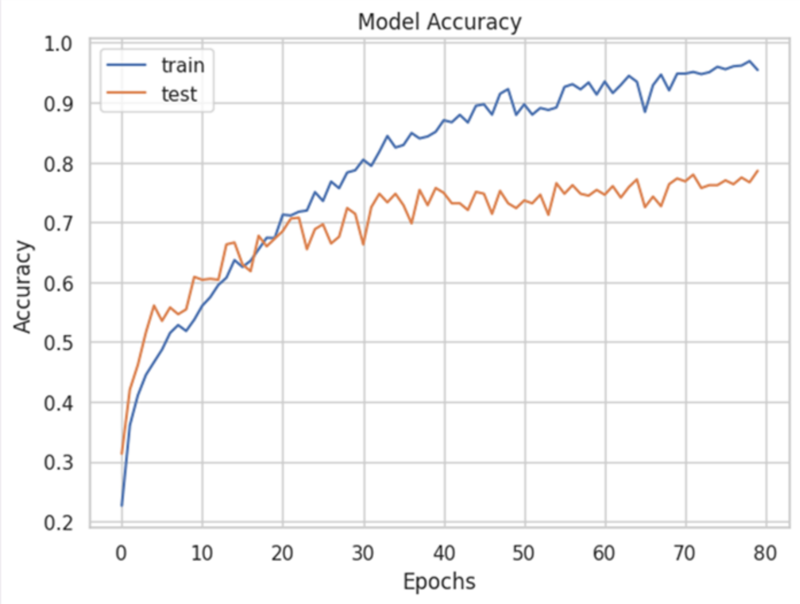
Accuracy : 76





Accuracy : 79





Accuracy : 76

Confusion Matrix

Actual \ Predicted	Bougainvillea	Daffodil	Dahlia	Foxglove	Hibiscus	Hydrangea	Orchid	Rose	Sunflower	Tulip
Bougainvillea	56	0	2	9	2	4	1	6	1	4
Daffodil	1	46	1	0	2	0	4	2	5	4
Dahlia	4	0	41	1	3	3	3	2	0	0
Foxglove	2	1	0	69	0	2	2	0	0	0
Hibiscus	4	2	2	0	52	0	2	2	0	1
Hydrangea	3	0	1	3	0	34	3	0	0	2
Orchid	6	0	0	2	4	2	42	0	1	2
Rose	5	0	1	2	3	1	3	42	0	5
Sunflower	0	0	0	0	0	0	0	0	56	0
Tulip	2	1	1	1	1	1	3	4	1	40

	precision	recall	f1-score	support
0	0.66	0.67	0.67	83
1	0.71	0.92	0.80	50
2	0.72	0.84	0.77	49
3	0.91	0.79	0.85	87
4	0.80	0.78	0.79	67
5	0.74	0.72	0.73	47
6	0.71	0.67	0.69	63
7	0.68	0.72	0.70	58
8	1.00	0.88	0.93	64
9	0.73	0.69	0.71	58
accuracy			0.76	626
macro avg	0.76	0.77	0.76	626
weighted avg	0.77	0.76	0.76	626

Accuracy : 79

Confusion Matrix

Actual \ Predicted	Bougainvillea	Daffodil	Dahlia	Foxglove	Hibiscus	Hydrangea	Orchid	Rose	Sunflower	Tulip
Bougainvillea	68	0	1	1	1	1	0	1	0	1
Daffodil	0	39	1	0	3	0	0	0	20	3
Dahlia	0	0	56	0	0	1	1	2	0	0
Foxglove	1	0	0	57	0	3	1	2	0	1
Hibiscus	3	0	2	0	52	0	2	2	0	0
Hydrangea	2	1	6	0	0	33	5	0	1	2
Orchid	4	1	9	0	1	3	45	1	0	0
Rose	10	1	9	0	3	0	2	26	1	3
Sunflower	0	1	1	0	0	0	0	0	65	0
Tulip	3	1	0	0	2	0	0	3	0	55

	precision	recall	f1-score	support
0	0.92	0.75	0.82	91
1	0.59	0.89	0.71	44
2	0.93	0.66	0.77	85
3	0.88	0.98	0.93	58
4	0.85	0.84	0.85	62
5	0.66	0.80	0.73	41
6	0.70	0.80	0.75	56
7	0.47	0.70	0.57	37
8	0.97	0.75	0.84	87
9	0.86	0.85	0.85	65
accuracy			0.79	626
macro avg	0.78	0.80	0.78	626
weighted avg	0.83	0.79	0.80	626

Actual : Rose
Predicted: Orchid

Actual : Rose
Predicted : Rose

Actual : Dahlia
Predicted : Tulip

Actual : Hydrangea
Predicted : Hydrangea

Accuracy : 76

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 256, 256, 32)	2432
max_pooling2d (MaxPooling2D)	(None, 128, 128, 32)	0
conv2d_1 (Conv2D)	(None, 128, 128, 64)	18496
max_pooling2d_1 (MaxPooling 2D)	(None, 64, 64, 64)	0
conv2d_2 (Conv2D)	(None, 64, 64, 96)	55392
max_pooling2d_2 (MaxPooling 2D)	(None, 32, 32, 96)	0
conv2d_3 (Conv2D)	(None, 32, 32, 96)	83040
max_pooling2d_3 (MaxPooling 2D)	(None, 16, 16, 96)	0
conv2d_4 (Conv2D)	(None, 16, 16, 96)	83040
max_pooling2d_4 (MaxPooling 2D)	(None, 8, 8, 96)	0
flatten (Flatten)	(None, 6144)	0
dense (Dense)	(None, 512)	3146240
dense_1 (Dense)	(None, 10)	5130

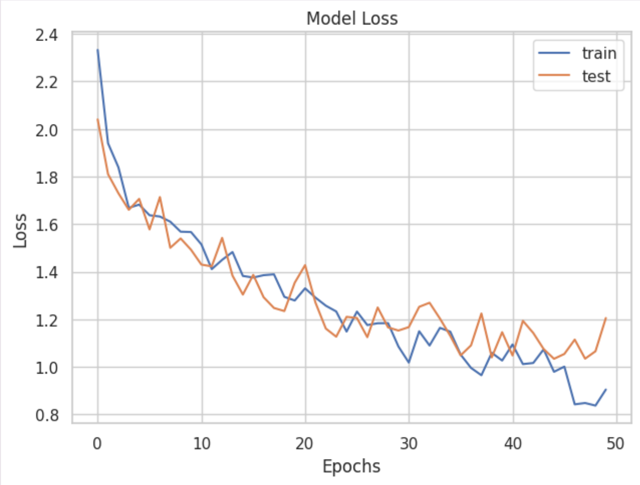
Total params: 3,393,770
Trainable params: 3,393,770
Non-trainable params: 0

Accuracy : 79

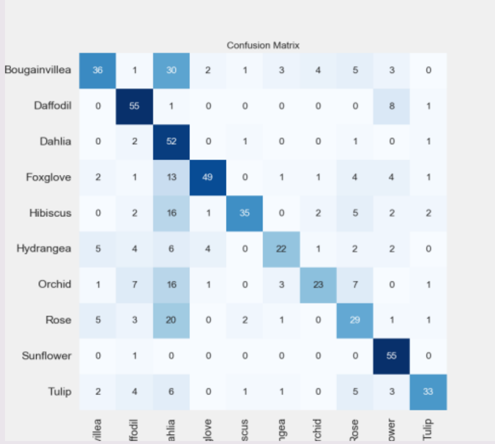
Layer (type)	Output Shape	Param #
conv2d_10 (Conv2D)	(None, 150, 150, 32)	2432
max_pooling2d_10 (MaxPoolin g2D)	(None, 75, 75, 32)	0
conv2d_11 (Conv2D)	(None, 75, 75, 64)	18496
max_pooling2d_11 (MaxPoolin g2D)	(None, 37, 37, 64)	0
conv2d_12 (Conv2D)	(None, 37, 37, 96)	55392
max_pooling2d_12 (MaxPoolin g2D)	(None, 18, 18, 96)	0
conv2d_13 (Conv2D)	(None, 18, 18, 96)	83040
max_pooling2d_13 (MaxPoolin g2D)	(None, 9, 9, 96)	0
flatten_2 (Flatten)	(None, 7776)	0
dense_4 (Dense)	(None, 512)	3981824
activation (Activation)	(None, 512)	0
dense_5 (Dense)	(None, 10)	5130

Total params: 4,146,314
Trainable params: 4,146,314
Non-trainable params: 0

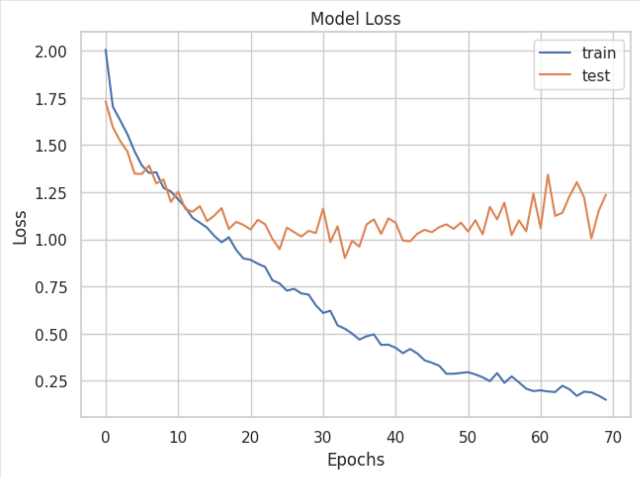
For Accuracy 62



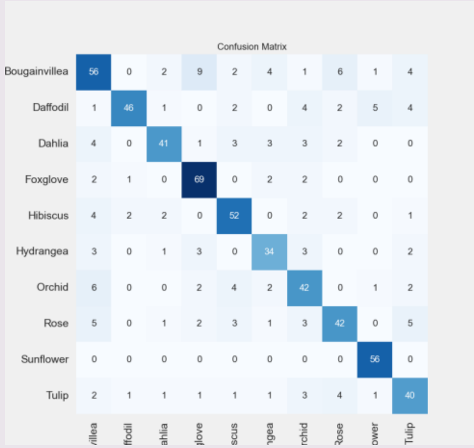
	precision	recall	f1-score	support
0	0.42	0.71	0.53	51
1	0.85	0.69	0.76	80
2	0.91	0.33	0.48	160
3	0.64	0.86	0.74	57
4	0.54	0.88	0.67	40
5	0.48	0.71	0.57	31
6	0.39	0.74	0.51	31
7	0.47	0.50	0.48	58
8	0.98	0.71	0.82	78
9	0.60	0.82	0.69	40
accuracy			0.62	626
macro avg	0.63	0.69	0.63	626
weighted avg	0.72	0.62	0.62	626

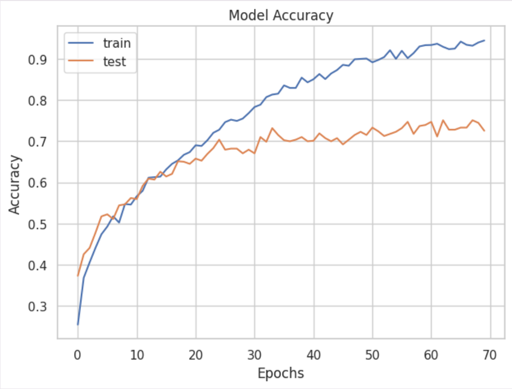


For Accuracy 70

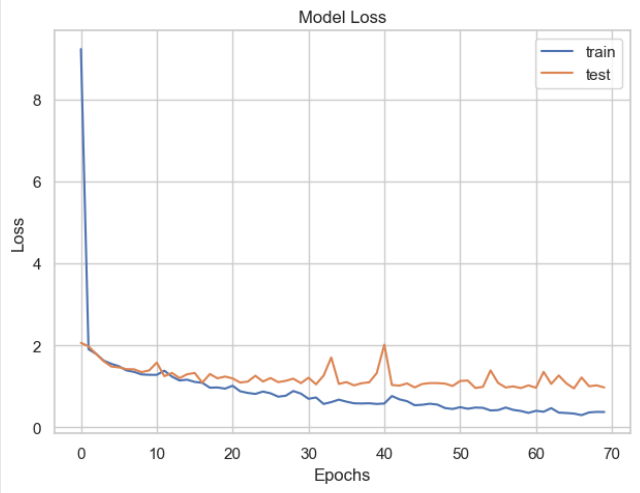


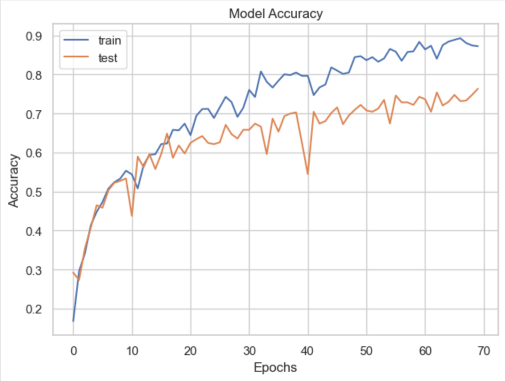
	precision	recall	f1-score	support
0	0.68	0.76	0.72	76
1	0.63	0.89	0.74	46
2	0.96	0.34	0.51	160
3	0.78	0.91	0.84	65
4	0.69	0.83	0.76	54
5	0.50	0.85	0.63	27
6	0.41	0.80	0.54	30
7	0.60	0.71	0.65	52
8	0.95	0.79	0.86	67
9	0.82	0.92	0.87	49
accuracy			0.70	626
macro avg	0.70	0.78	0.71	626
weighted avg	0.77	0.70	0.69	626



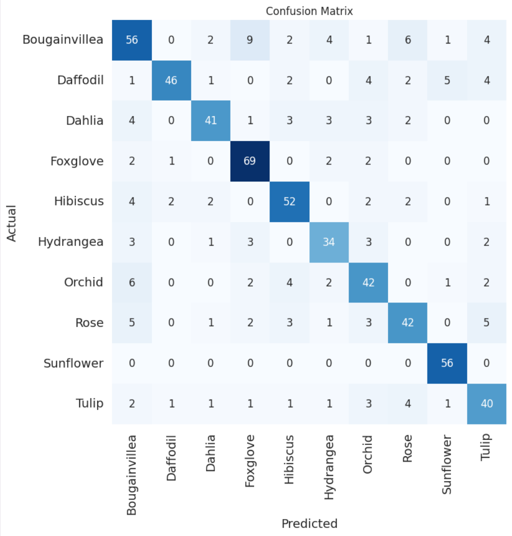


For Accuracy 76

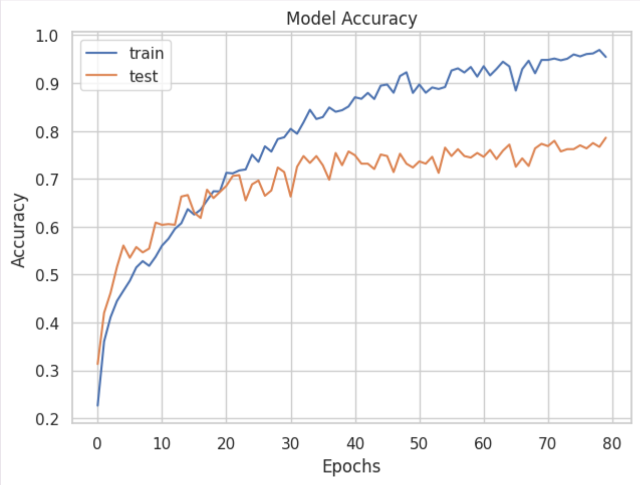


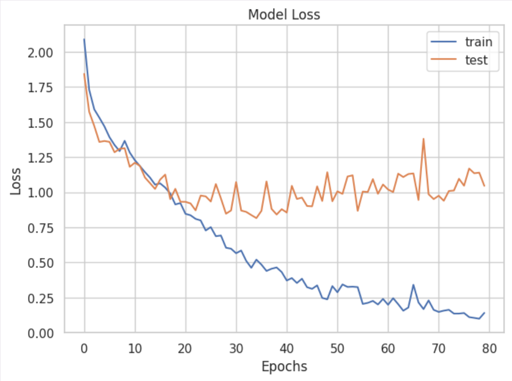


	precision	recall	f1-score	support
0	0.66	0.67	0.67	83
1	0.71	0.92	0.80	50
2	0.72	0.84	0.77	49
3	0.91	0.79	0.85	87
4	0.80	0.78	0.79	67
5	0.74	0.72	0.73	47
6	0.71	0.67	0.69	63
7	0.68	0.72	0.70	58
8	1.00	0.88	0.93	64
9	0.73	0.69	0.71	58
accuracy			0.76	626
macro avg	0.76	0.77	0.76	626
weighted avg	0.77	0.76	0.76	626

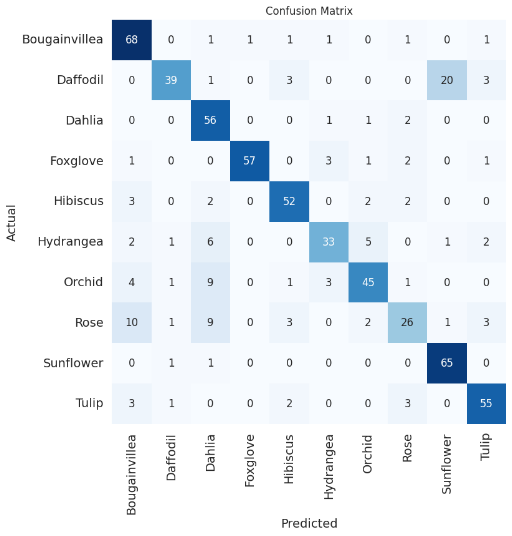


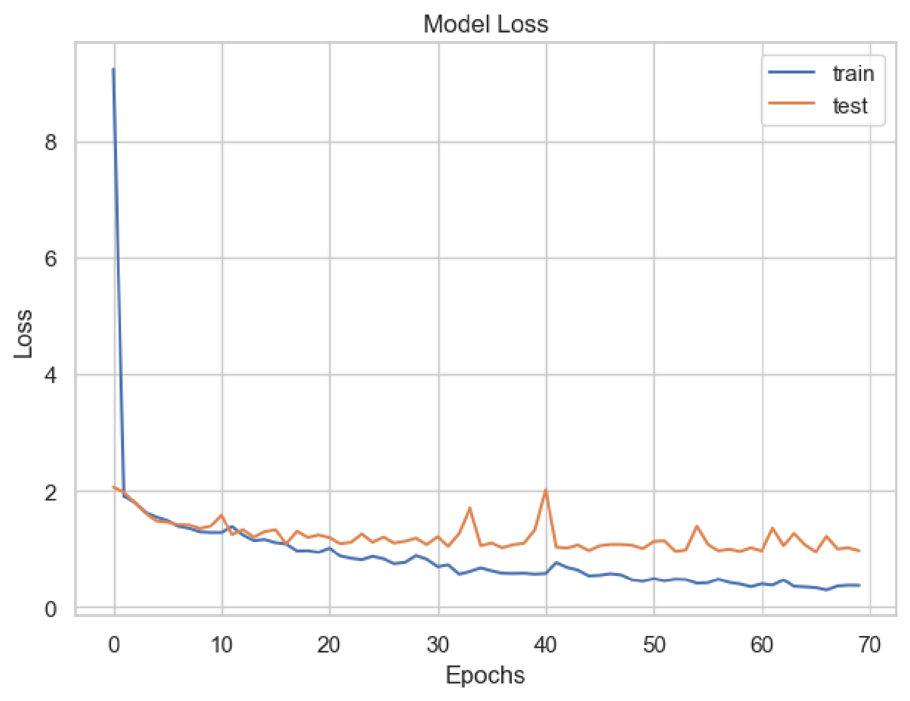
For Accuracy 79

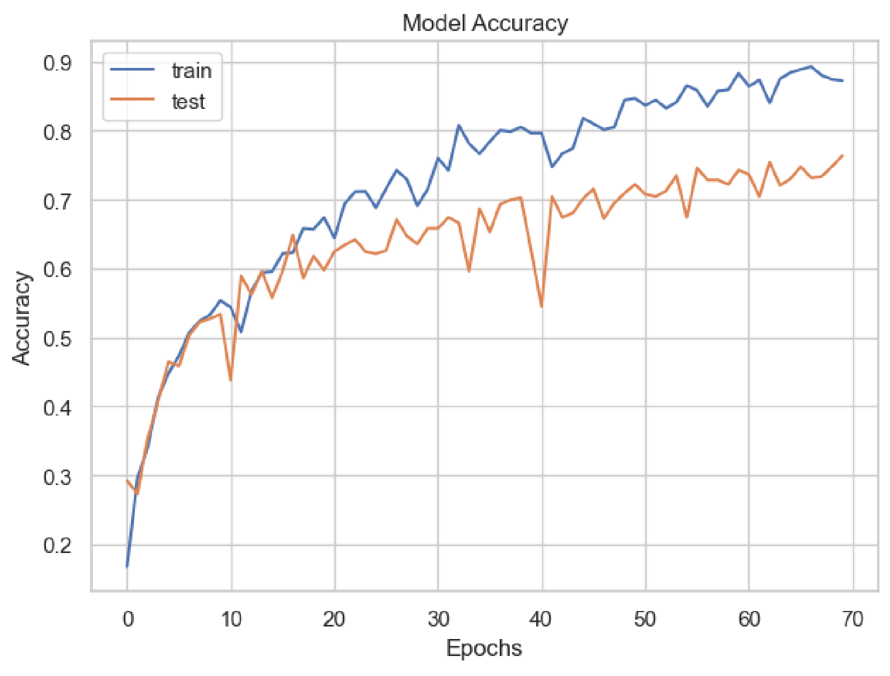


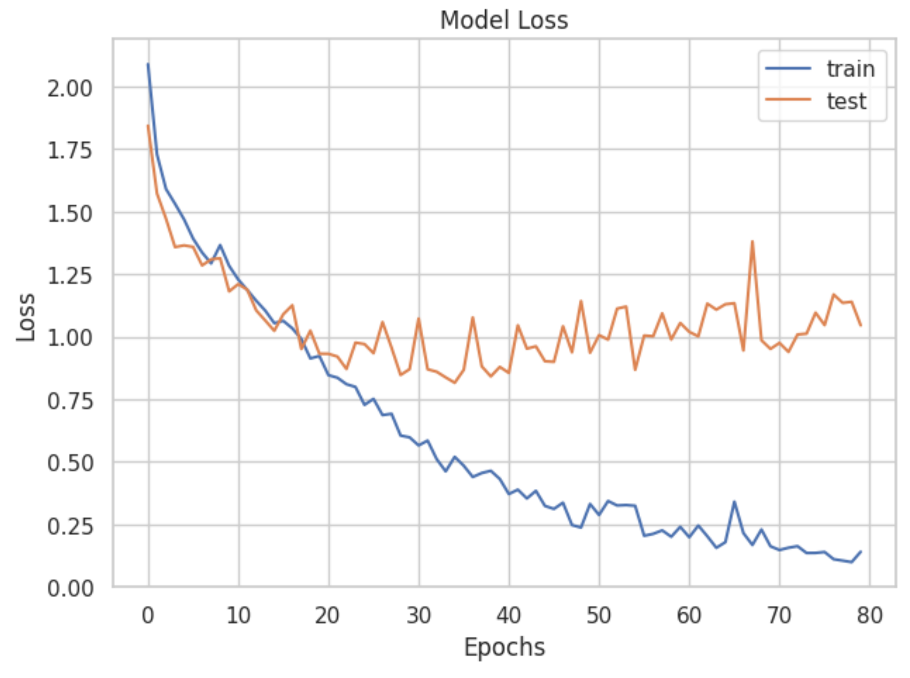


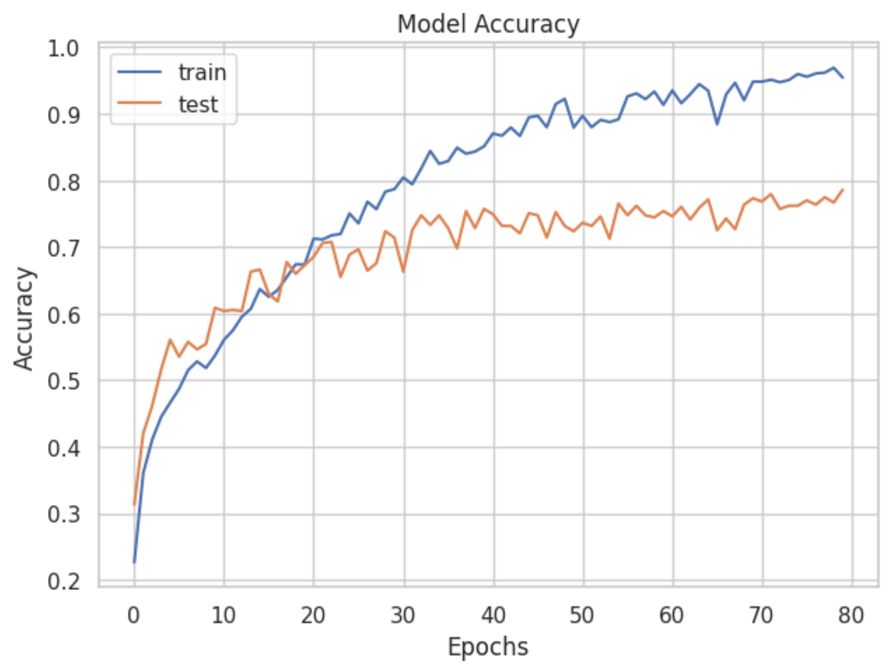
	precision	recall	f1-score	support
0	0.92	0.75	0.82	91
1	0.59	0.89	0.71	44
2	0.93	0.66	0.77	85
3	0.88	0.98	0.93	58
4	0.85	0.84	0.85	62
5	0.66	0.80	0.73	41
6	0.70	0.80	0.75	56
7	0.47	0.70	0.57	37
8	0.97	0.75	0.84	87
9	0.86	0.85	0.85	65
accuracy			0.79	626
macro avg	0.78	0.80	0.78	626
weighted avg	0.83	0.79	0.80	626

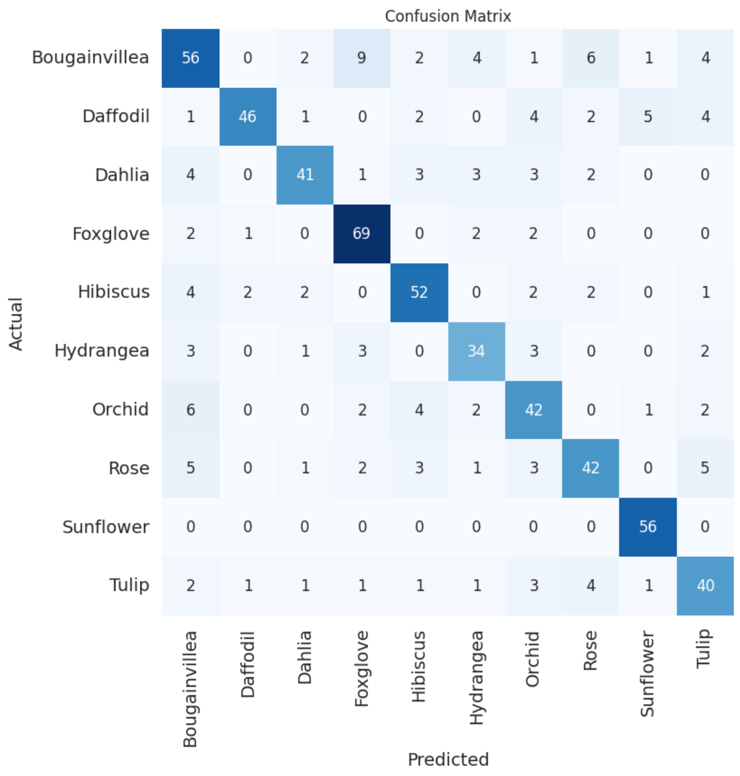



Model Loss
train
test
Loss
Epochs

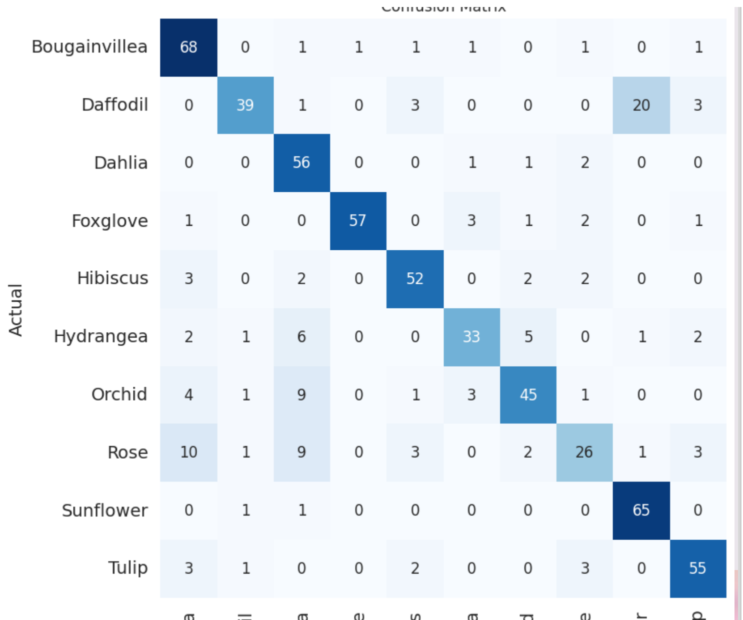

Model Accuracy
train
test
Accuracy
Epochs


Model Loss
train
test
Loss
Epochs


Model Accuracy
train
test
Accuracy
Epochs



	precision	recall	f1-score	support
0	0.66	0.67	0.67	83
1	0.71	0.92	0.80	50
2	0.72	0.84	0.77	49
3	0.91	0.79	0.85	87
4	0.80	0.78	0.79	67
5	0.74	0.72	0.73	47
6	0.71	0.67	0.69	63
7	0.68	0.72	0.70	58
8	1.00	0.88	0.93	64
9	0.73	0.69	0.71	58
accuracy			0.76	626
macro avg	0.76	0.77	0.76	626
weighted avg	0.77	0.76	0.76	626



	precision	recall	f1-score	support
0	0.92	0.75	0.82	91
1	0.59	0.89	0.71	44
2	0.93	0.66	0.77	85
3	0.88	0.98	0.93	58
4	0.85	0.84	0.85	62
5	0.66	0.80	0.73	41
6	0.70	0.80	0.75	56
7	0.47	0.70	0.57	37
8	0.97	0.75	0.84	87
9	0.86	0.85	0.85	65
accuracy			0.79	626