Dataset and Dataset Preprocessing:
I used a flower dataset. The dataset contained five types of flowers (dandelions, roses, daisies, sunflowers, tulips). I split the dataset into three parts: for training, I used 70% of the dataset; for validation, I used 15% of the dataset; and for testing, I used 15% of the dataset. There were a total of 3671 images.

- Number of training examples: 2568
- Number of validation examples: 551
- Number of testing examples: 552

I applied standard data preprocessing included in the repository, which involves applying random resized crop to the dataset.

My Model:

```
BagNet(
  (conv1): Conv2d(3, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (conv2): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (conv3): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (conv4): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (fc1): Linear(in_features=12544, out_features=128, bias=True)
  (fc2): Linear(in_features=128, out_features=num_classes, bias=True)
)
```

Layer (type (var_name))	Input Shape	Output Shape	Param #	Trainable
BagNet (BagNet)	[batch_size, 3, 224, 224]	[batch_size, num_classes]	--	True
├─Conv2d (conv1)	[batch_size, 3, 224, 224]	[batch_size, 16, 224, 224]	448	True
├─MaxPool2d (pool)	[batch_size, 16, 224, 224]	[batch_size, 16, 112, 112]	--	--
├─Conv2d (conv2)	[batch_size, 16, 112, 112]	[batch_size, 32, 112, 112]	4,640	True
├─MaxPool2d (pool)	[batch_size, 32, 112, 112]	[batch_size, 32, 56, 56]	--	--
├─Conv2d (conv3)	[batch_size, 32, 56, 56]	[batch_size, 64, 56, 56]	18,496	True
├─MaxPool2d (pool)	[batch_size, 64, 56, 56]	[batch_size, 64, 28, 28]	--	--
├─Conv2d (conv4)	[batch_size, 64, 28, 28]	[batch_size, 64, 28, 28]	36,928	True
├─MaxPool2d (pool)	[batch_size, 64, 28, 28]	[batch_size, 64, 14, 14]	--	--
├─Linear (fc1)	[batch_size, 64*14*14]	[batch_size, 128]	1,179,776	True

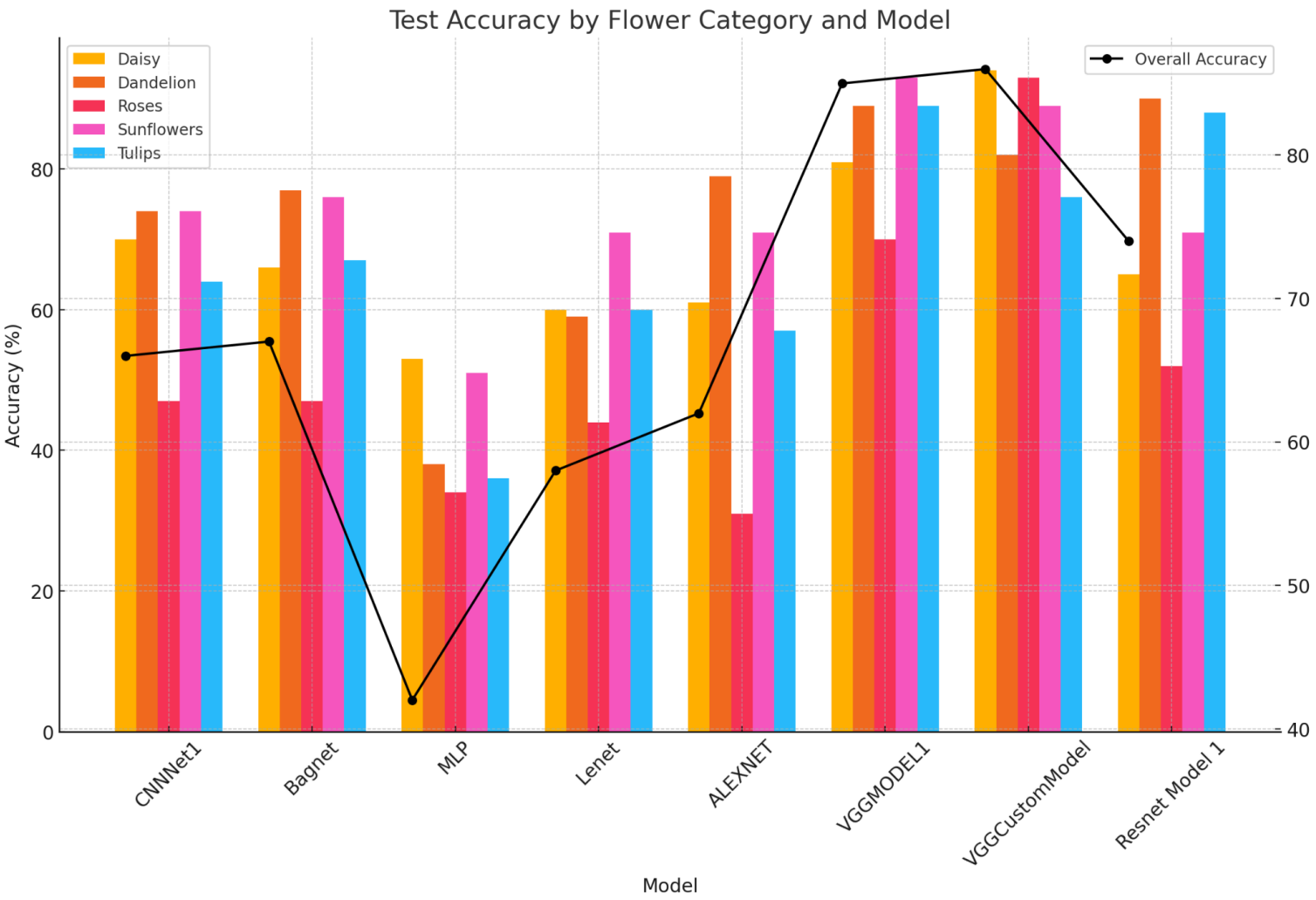
├─Linear (fc2) [batch_size, 128] [batch_size, num_classes] 129
True
==========
Total params: 1,240,417
Trainable params: 1,240,417
Non-trainable params: 0

Training:
For all the models I tested, I kept the following hyperparameters the same:

- Image Height: 224
- Image Width: 224
    - Learning Rate Schedule: StepLR
    - Learning Rate: 0.1
    - Momentum: 0.9
    - Weight Decay: 1e-4
- Optimizer: Adam
- Epochs: 40

Testing Results:
The results highlighted in green represent my model, BagNet, while the result highlighted in orange indicates the best performance achieved.

Model	Daisy	Dandelion	Roses	Sunflowers	Tulips	Overall Accuracy
CNNNet1	70% (59/84)	74% (98/131)	47% (44/92)	74% (72/97)	64% (70/108)	66% (343/512)
Bagnet	66% (54/81)	77% (98/127)	47% (44/93)	76% (74/97)	67% (77/114)	67% (347/512)
MLP	53% (49/91)	38% (49/127)	34% (37/108)	51% (47/91)	36% (35/96)	42% (217/513)
Lenet	60% (60/100)	59% (72/122)	44% (44/98)	71% (52/73)	60% (72/119)	58% (300/512)
ALEXNET	61% (44/71)	79% (107/134)	31% (27/85)	71% (84/118)	57% (60/104)	62% (322/512)
VGGMODEL 1	81% (78/96)	89% (114/128)	70% (65/92)	93% (91/97)	89% (89/99)	85% (437/512)
VGGCustom Model	94% (75/79)	82% (92/111)	93% (98/105)	89% (88/98)	76% (91/119)	86% (444/512)
Resnet Model 1	65% (55/84)	90% (120/133)	52% (57/108)	71% (65/91)	88% (85/96)	74% (382/512)


Test Accuracy by Flower Category and Model
Daisy
Dandelion
Roses
Sunflowers
Tulips
Overall Accuracy
Accuracy (%)
0
20
40
60
80
40
50
60
70
80
CNNNet1
Bagnet
MLP
Lenet
ALEXNET
VGGMODEL1
VGGCustomModel
Resnet Model 1
Model