

```

In [8]: from os import listdir
        from os.path import join, isfile, isdir
        import pandas as pd

        from ImageClassification_marlog_acc import ImageClassification

        from keras.models import Sequential
        from keras.layers import Activation, Dropout, Flatten, Dense
        from keras.optimizers import Adam

        # Warnings about data rate limits
        # IOPub data rate exceeded.
        # https://github.com/jupyter/notebook/issues/2287
        # --NotebookApp.iopub_data_rate_limit (increase limit)
        # jupyter notebook --generate-config

In [9]: species_home_directory = '/home/marlog/Rprojects/WildflowersAll104May18/'
        training_flag = 'Training'
        validation_flag = 'Validation'
        # List all the directories, these become the classes
        species_directories = [dir for dir in listdir(species_home_directory) if isdir(join(species_home_directory, dir))]

        # Create the base dataframe
        wildflower_df = pd.DataFrame()
        validation_df = pd.DataFrame()

        for current_species in species_directories:
            # Get the training images
            images = [f for f in listdir(join(species_home_directory, current_species, training_flag))
                      if isfile(join(species_home_directory, current_species, training_flag, f))]
            # Only for cpu testing
            images = images[0:5000]
            species_class = [current_species] * len(images)
            species_directory = [join(species_home_directory, current_species, training_flag)] * len(images)
            dict_rep = {'image_id': images, 'directory': species_directory, 'class': species_class}
            df_rep = pd.DataFrame(dict_rep)
            wildflower_df = wildflower_df.append(df_rep, ignore_index=True)

        for current_species in species_directories:
            # Get the validation images
            images = [f for f in listdir(join(species_home_directory, current_species, validation_flag))
                      if isfile(join(species_home_directory, current_species, validation_flag, f))]
            # Only for cpu testing
            images = images[0:5000]
            species_class = [current_species] * len(images)
            species_directory = [join(species_home_directory, current_species, validation_flag)] * len(images)
            dict_rep = {'image_id': images, 'directory': species_directory, 'class': species_class}
            df_rep = pd.DataFrame(dict_rep)
            validation_df = validation_df.append(df_rep, ignore_index=True)

        print("Length of train/test data: {}, length of validation data: {}".format(wildflower_df.shape[0], validation_df.shape[0]))

        Length of train/test data: 23756, length of validation data: 5927

In [10]: wildflower_classifier = ImageClassification('id1',
                                                    wildflower_df,
                                                    validation_df,
                                                    '/home/marlog/Rprojects/Wildflowers04May18',
                                                    32,
                                                    0.25,
                                                    1,
                                                    True,
                                                    True)

```

(<http://localhost:8888/>) Successfully loaded.

```
In [11]: inception_resnet_top_model_basic = Sequential()
inception_resnet_top_model_basic.add(Dense(wildflower_classifier.class_counts.shape[0], activation='softmax', input_shape=(wildflower_classifier.model_dense_pooled_dimensions.get('inception-resnet'))))
inception_resnet_top_model_basic.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

model_dict = {'BasicV3Res': [inception_resnet_top_model_basic, 'inception-resnet', 500]}

wildflower_classifier.train_provided_top_models(model_dict)
```

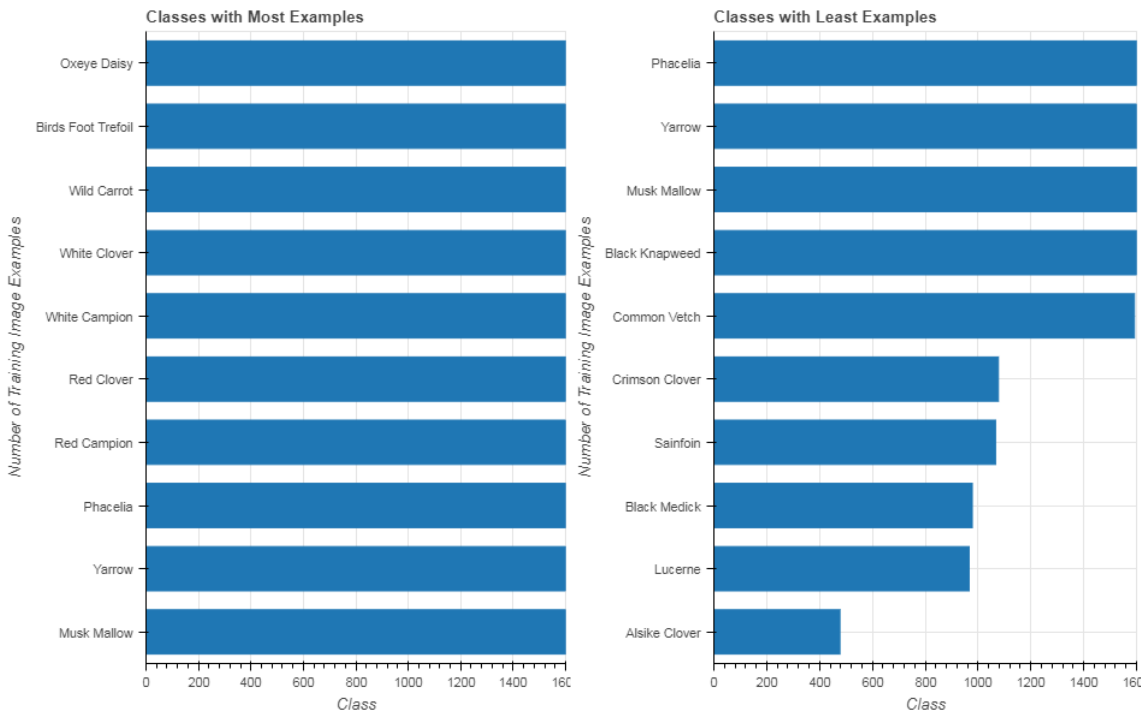
Beginning to create the train bottleneck features for model: inception-resnet
Found 17818 images belonging to 17 classes.
Beginning to create the test bottleneck features for model: inception-resnet
Found 5938 images belonging to 17 classes.
Beginning to train model: BasicV3Res for 500 epochs.
Epoch 1/500
557/557 [=====] - 4s 6ms/step - loss: 1.6956 - acc: 0.5653 - val_loss: 1.0688 - val_acc: 0.6598
Epoch 2/500
557/557 [=====] - 3s 5ms/step - loss: 1.1883 - acc: 0.6945 - val_loss: 0.9449 - val_acc: 0.6952
Epoch 3/500
557/557 [=====] - 3s 5ms/step - loss: 1.0491 - acc: 0.7265 - val_loss: 0.8777 - val_acc: 0.7189
Epoch 4/500
557/557 [=====] - 3s 5ms/step - loss: 0.9658 - acc: 0.7462 - val_loss: 0.8408 - val_acc: 0.7309
Epoch 5/500
557/557 [=====] - 3s 5ms/step - loss: 0.9076 - acc: 0.7602 - val_loss: 0.8156 - val_acc: 0.7412
Epoch 6/500
557/557 [=====] - 3s 5ms/step - loss: 0.8613 - acc: 0.7738 - val_loss: 0.7917 - val_acc: 0.7489
Epoch 7/500
557/557 [=====] - 3s 5ms/step - loss: 0.8242 - acc: 0.7816 - val_loss: 0.7729 - val_acc: 0.7587
Epoch 8/500
557/557 [=====] - 3s 5ms/step - loss: 0.7931 - acc: 0.7899 - val_loss: 0.7527 - val_acc: 0.7661
Epoch 9/500
557/557 [=====] - 3s 5ms/step - loss: 0.7654 - acc: 0.7972 - val_loss: 0.7367 - val_acc: 0.7747
Epoch 10/500
557/557 [=====] - 3s 5ms/step - loss: 0.7420 - acc: 0.8037 - val_loss: 0.7259 - val_acc: 0.7784
Epoch 11/500
557/557 [=====] - 3s 5ms/step - loss: 0.7200 - acc: 0.8094 - val_loss: 0.7168 - val_acc: 0.7824
Epoch 12/500
557/557 [=====] - 3s 5ms/step - loss: 0.7010 - acc: 0.8139 - val_loss: 0.7060 - val_acc: 0.7868
Epoch 13/500
557/557 [=====] - 3s 5ms/step - loss: 0.6825 - acc: 0.8186 - val_loss: 0.7028 - val_acc: 0.7878
Epoch 14/500
557/557 [=====] - 3s 5ms/step - loss: 0.6659 - acc: 0.8232 - val_loss: 0.6956 - val_acc: 0.7913
Epoch 15/500
557/557 [=====] - 3s 5ms/step - loss: 0.6506 - acc: 0.8275 - val_loss: 0.6932 - val_acc: 0.7940
Epoch 16/500
557/557 [=====] - 3s 5ms/step - loss: 0.6373 - acc: 0.8317 - val_loss: 0.6889 - val_acc: 0.7940
Epoch 17/500
557/557 [=====] - 3s 5ms/step - loss: 0.6236 - acc: 0.8359 - val_loss: 0.6881 - val_acc: 0.7947
Epoch 18/500
557/557 [=====] - 3s 5ms/step - loss: 0.6119 - acc: 0.8391 - val_loss: 0.6837 - val_acc: 0.7976
Epoch 19/500
557/557 [=====] - 3s 5ms/step - loss: 0.5997 - acc: 0.8429 - val_loss: 0.6823 - val_acc: 0.7977
Epoch 20/500
557/557 [=====] - 3s 5ms/step - loss: 0.5888 - acc: 0.8460 - val_loss: 0.6823 - val_acc: 0.8006
Epoch 21/500
557/557 [=====] - 3s 5ms/step - loss: 0.5782 - acc: 0.8485 - val_loss: 0.6832 - val_acc: 0.8004
Epoch 22/500
557/557 [=====] - 3s 5ms/step - loss: 0.5677 - acc: 0.8524 - val_loss: 0.6851 - val_acc: 0.8016
Epoch 23/500
557/557 [=====] - 3s 5ms/step - loss: 0.5584 - acc: 0.8555 - val_loss: 0.6842 - val_acc: 0.8040
Epoch 24/500
557/557 [=====] - 3s 5ms/step - loss: 0.5494 - acc: 0.8579 - val_loss: 0.6828 - val_acc: 0.8040
Epoch 25/500
557/557 [=====] - 3s 5ms/step - loss: 0.5405 - acc: 0.8603 - val_loss: 0.6823 - val_acc: 0.8048
Epoch 26/500
557/557 [=====] - 3s 5ms/step - loss: 0.5326 - acc: 0.8632 - val_loss: 0.6808 - val_acc: 0.8045
Epoch 27/500
557/557 [=====] - 3s 5ms/step - loss: 0.5247 - acc: 0.8651 - val_loss: 0.6784 - val_acc: 0.8028
Epoch 28/500
557/557 [=====] - 3s 5ms/step - loss: 0.5162 - acc: 0.8673 - val_loss: 0.6803 - val_acc: 0.8018
Epoch 29/500
557/557 [=====] - 3s 5ms/step - loss: 0.5097 - acc: 0.8694 - val_loss: 0.6806 - val_acc: 0.8009
Epoch 30/500
557/557 [=====] - 3s 5ms/step - loss: 0.5020 - acc: 0.8718 - val_loss: 0.6875 - val_acc: 0.8004
Epoch 31/500
557/557 [=====] - 3s 5ms/step - loss: 0.4958 - acc: 0.8728 - val_loss: 0.6847 - val_acc: 0.8021
Epoch 32/500
557/557 [=====] - 3s 5ms/step - loss: 0.4889 - acc: 0.8746 - val_loss: 0.6840 - val_acc: 0.8040
Epoch 33/500
557/557 [=====] - 3s 5ms/step - loss: 0.4824 - acc: 0.8761 - val_loss: 0.6797 - val_acc: 0.8057
Epoch 34/500
557/557 [=====] - 3s 5ms/step - loss: 0.4758 - acc: 0.8788 - val_loss: 0.6735 - val_acc: 0.8080
Epoch 35/500
557/557 [=====] - 3s 5ms/step - loss: 0.4704 - acc: 0.8809 - val_loss: 0.6700 - val_acc: 0.8078
Epoch 36/500
557/557 [=====] - 3s 5ms/step - loss: 0.4643 - acc: 0.8830 - val_loss: 0.6655 - val_acc: 0.8107
Epoch 37/500
557/557 [=====] - 3s 5ms/step - loss: 0.4586 - acc: 0.8842 - val_loss: 0.6607 - val_acc: 0.8136
Epoch 38/500
557/557 [=====] - 3s 6ms/step - loss: 0.4529 - acc: 0.8856 - val_loss: 0.6556 - val_acc: 0.8144
Epoch 39/500
557/557 [=====] - 3s 5ms/step - loss: 0.4471 - acc: 0.8874 - val_loss: 0.6537 - val_acc: 0.8159
Epoch 40/500
557/557 [=====] - 3s 5ms/step - loss: 0.4419 - acc: 0.8888 - val_loss: 0.6544 - val_acc: 0.8153
Epoch 41/500
557/557 [=====] - 3s 5ms/step - loss: 0.4374 - acc: 0.8911 - val_loss: 0.6543 - val_acc: 0.8144
Epoch 42/500
557/557 [=====] - 3s 5ms/step - loss: 0.4320 - acc: 0.8931 - val_loss: 0.6608 - val_acc: 0.8139
Epoch 43/500
557/557 [=====] - 3s 5ms/step - loss: 0.4270 - acc: 0.8941 - val_loss: 0.6613 - val_acc: 0.8149
Epoch 44/500
557/557 [=====] - 3s 5ms/step - loss: 0.4228 - acc: 0.8950 - val_loss: 0.6648 - val_acc: 0.8154
Epoch 45/500
557/557 [=====] - 3s 5ms/step - loss: 0.4182 - acc: 0.8959 - val_loss: 0.6662 - val_acc: 0.8146
Epoch 46/500
557/557 [=====] - 3s 5ms/step - loss: 0.4131 - acc: 0.8975 - val_loss: 0.6691 - val_acc: 0.8142
Epoch 47/500
557/557 [=====] - 3s 5ms/step - loss: 0.4091 - acc: 0.8985 - val_loss: 0.6722 - val_acc: 0.8122
Epoch 48/500
557/557 [=====] - 3s 5ms/step - loss: 0.4044 - acc: 0.8999 - val_loss: 0.6753 - val_acc: 0.8124
Epoch 49/500

```
557/557 [=====] - 3s 5ms/step - loss: 0.4005 - acc: 0.9012 - val_loss: 0.6784 - val_acc: 0.8116
Epoch 50/500
557/557 [=====] - 3s 5ms/step - loss: 0.3966 - acc: 0.9022 - val_loss: 0.6781 - val_acc: 0.8114
Epoch 51/500
557/557 [=====] - 3s 5ms/step - loss: 0.3922 - acc: 0.9034 - val_loss: 0.6759 - val_acc: 0.8134
Epoch 52/500
557/557 [=====] - 3s 6ms/step - loss: 0.3885 - acc: 0.9048 - val_loss: 0.6771 - val_acc: 0.8154
Epoch 53/500
557/557 [=====] - 3s 5ms/step - loss: 0.3844 - acc: 0.9062 - val_loss: 0.6731 - val_acc: 0.8168
Epoch 54/500
557/557 [=====] - 3s 5ms/step - loss: 0.3807 - acc: 0.9068 - val_loss: 0.6696 - val_acc: 0.8169
Epoch 55/500
557/557 [=====] - 3s 5ms/step - loss: 0.3764 - acc: 0.9080 - val_loss: 0.6703 - val_acc: 0.8183
Epoch 56/500
557/557 [=====] - 3s 5ms/step - loss: 0.3730 - acc: 0.9088 - val_loss: 0.6683 - val_acc: 0.8191
Epoch 57/500
557/557 [=====] - 3s 5ms/step - loss: 0.3699 - acc: 0.9103 - val_loss: 0.6683 - val_acc: 0.8201
Epoch 58/500
557/557 [=====] - 3s 5ms/step - loss: 0.3660 - acc: 0.9112 - val_loss: 0.6681 - val_acc: 0.8210
Epoch 59/500
557/557 [=====] - 3s 5ms/step - loss: 0.3624 - acc: 0.9124 - val_loss: 0.6692 - val_acc: 0.8208
Epoch 60/500
557/557 [=====] - 3s 5ms/step - loss: 0.3590 - acc: 0.9136 - val_loss: 0.6691 - val_acc: 0.8198
Epoch 61/500
557/557 [=====] - 3s 5ms/step - loss: 0.3556 - acc: 0.9149 - val_loss: 0.6732 - val_acc: 0.8188
Epoch 62/500
557/557 [=====] - 3s 5ms/step - loss: 0.3528 - acc: 0.9157 - val_loss: 0.6709 - val_acc: 0.8193
Epoch 63/500
557/557 [=====] - 3s 5ms/step - loss: 0.3496 - acc: 0.9167 - val_loss: 0.6727 - val_acc: 0.8188
Epoch 64/500
557/557 [=====] - 3s 5ms/step - loss: 0.3462 - acc: 0.9176 - val_loss: 0.6744 - val_acc: 0.8190
Epoch 65/500
557/557 [=====] - 3s 5ms/step - loss: 0.3428 - acc: 0.9183 - val_loss: 0.6776 - val_acc: 0.8185
Epoch 66/500
557/557 [=====] - 3s 5ms/step - loss: 0.3399 - acc: 0.9189 - val_loss: 0.6745 - val_acc: 0.8191
Epoch 67/500
557/557 [=====] - 3s 5ms/step - loss: 0.3372 - acc: 0.9196 - val_loss: 0.6776 - val_acc: 0.8183
Epoch 68/500
557/557 [=====] - 3s 5ms/step - loss: 0.3341 - acc: 0.9205 - val_loss: 0.6767 - val_acc: 0.8210
Epoch 69/500
557/557 [=====] - 3s 5ms/step - loss: 0.3309 - acc: 0.9214 - val_loss: 0.6750 - val_acc: 0.8232
Epoch 70/500
557/557 [=====] - 3s 5ms/step - loss: 0.3278 - acc: 0.9223 - val_loss: 0.6772 - val_acc: 0.8232
Epoch 71/500
557/557 [=====] - 3s 5ms/step - loss: 0.3249 - acc: 0.9230 - val_loss: 0.6798 - val_acc: 0.8227
Epoch 72/500
557/557 [=====] - 3s 5ms/step - loss: 0.3221 - acc: 0.9237 - val_loss: 0.6779 - val_acc: 0.8218
Epoch 73/500
557/557 [=====] - 3s 5ms/step - loss: 0.3199 - acc: 0.9241 - val_loss: 0.6869 - val_acc: 0.8201
Epoch 74/500
557/557 [=====] - 3s 5ms/step - loss: 0.3163 - acc: 0.9247 - val_loss: 0.6897 - val_acc: 0.8198
Epoch 75/500
557/557 [=====] - 3s 5ms/step - loss: 0.3146 - acc: 0.9255 - val_loss: 0.6945 - val_acc: 0.8190
Epoch 76/500
557/557 [=====] - 3s 5ms/step - loss: 0.3114 - acc: 0.9264 - val_loss: 0.6963 - val_acc: 0.8185
Epoch 77/500
557/557 [=====] - 3s 6ms/step - loss: 0.3093 - acc: 0.9275 - val_loss: 0.6992 - val_acc: 0.8200
Epoch 78/500
557/557 [=====] - 3s 5ms/step - loss: 0.3068 - acc: 0.9281 - val_loss: 0.6997 - val_acc: 0.8200
Epoch 79/500
557/557 [=====] - 3s 5ms/step - loss: 0.3043 - acc: 0.9291 - val_loss: 0.7017 - val_acc: 0.8183
Epoch 80/500
557/557 [=====] - 3s 5ms/step - loss: 0.3014 - acc: 0.9299 - val_loss: 0.7020 - val_acc: 0.8174
Epoch 81/500
557/557 [=====] - 3s 6ms/step - loss: 0.2991 - acc: 0.9306 - val_loss: 0.7015 - val_acc: 0.8178
Epoch 82/500
557/557 [=====] - 3s 5ms/step - loss: 0.2969 - acc: 0.9314 - val_loss: 0.7020 - val_acc: 0.8196
Epoch 83/500
557/557 [=====] - 3s 5ms/step - loss: 0.2945 - acc: 0.9321 - val_loss: 0.7037 - val_acc: 0.8173
Epoch 84/500
557/557 [=====] - 3s 5ms/step - loss: 0.2921 - acc: 0.9324 - val_loss: 0.7028 - val_acc: 0.8193
Epoch 85/500
557/557 [=====] - 3s 5ms/step - loss: 0.2900 - acc: 0.9332 - val_loss: 0.7017 - val_acc: 0.8205
Epoch 86/500
557/557 [=====] - 3s 5ms/step - loss: 0.2877 - acc: 0.9342 - val_loss: 0.7034 - val_acc: 0.8206
Epoch 87/500
557/557 [=====] - 3s 5ms/step - loss: 0.2854 - acc: 0.9348 - val_loss: 0.7070 - val_acc: 0.8174
Epoch 88/500
557/557 [=====] - 3s 5ms/step - loss: 0.2831 - acc: 0.9353 - val_loss: 0.7082 - val_acc: 0.8193
Epoch 89/500
557/557 [=====] - 3s 5ms/step - loss: 0.2811 - acc: 0.9358 - val_loss: 0.7104 - val_acc: 0.8193
Epoch 90/500
557/557 [=====] - 3s 5ms/step - loss: 0.2786 - acc: 0.9364 - val_loss: 0.7143 - val_acc: 0.8188
Epoch 91/500
557/557 [=====] - 3s 5ms/step - loss: 0.2764 - acc: 0.9369 - val_loss: 0.7142 - val_acc: 0.8188
Epoch 92/500
557/557 [=====] - 3s 5ms/step - loss: 0.2745 - acc: 0.9372 - val_loss: 0.7156 - val_acc: 0.8198
Epoch 93/500
557/557 [=====] - 3s 5ms/step - loss: 0.2721 - acc: 0.9375 - val_loss: 0.7129 - val_acc: 0.8215
Epoch 94/500
557/557 [=====] - 3s 5ms/step - loss: 0.2703 - acc: 0.9379 - val_loss: 0.7095 - val_acc: 0.8225
Epoch 95/500
557/557 [=====] - 3s 5ms/step - loss: 0.2685 - acc: 0.9384 - val_loss: 0.7028 - val_acc: 0.8245
Epoch 96/500
557/557 [=====] - 3s 5ms/step - loss: 0.2664 - acc: 0.9391 - val_loss: 0.6955 - val_acc: 0.8255
Epoch 97/500
557/557 [=====] - 3s 5ms/step - loss: 0.2644 - acc: 0.9398 - val_loss: 0.6975 - val_acc: 0.8258
Epoch 98/500
557/557 [=====] - 3s 5ms/step - loss: 0.2623 - acc: 0.9402 - val_loss: 0.6971 - val_acc: 0.8244
Epoch 99/500
557/557 [=====] - 3s 5ms/step - loss: 0.2601 - acc: 0.9407 - val_loss: 0.6928 - val_acc: 0.8267
Epoch 100/500
```

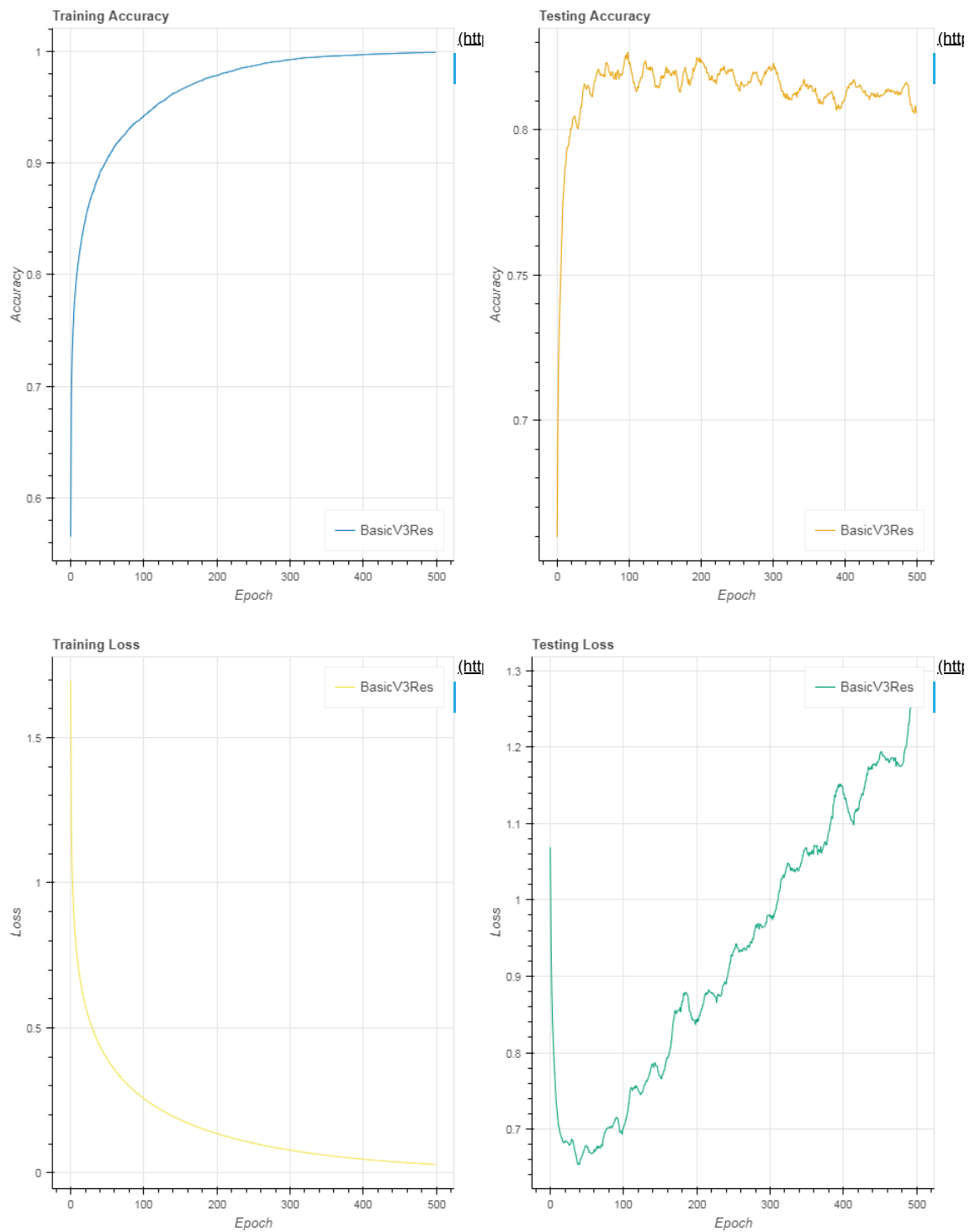
```
557/557 [=====] - 3s 5ms/step - loss: 0.0357 - acc: 0.9984 - val_loss: 1.1871 - val_acc: 0.8129
Epoch 458/500
557/557 [=====] - 3s 5ms/step - loss: 0.0355 - acc: 0.9984 - val_loss: 1.1858 - val_acc: 0.8139
Epoch 459/500
557/557 [=====] - 3s 5ms/step - loss: 0.0354 - acc: 0.9984 - val_loss: 1.1829 - val_acc: 0.8131
Epoch 460/500
557/557 [=====] - 3s 5ms/step - loss: 0.0352 - acc: 0.9984 - val_loss: 1.1823 - val_acc: 0.8139
Epoch 461/500
557/557 [=====] - 3s 5ms/step - loss: 0.0350 - acc: 0.9984 - val_loss: 1.1850 - val_acc: 0.8141
Epoch 462/500
557/557 [=====] - 3s 6ms/step - loss: 0.0349 - acc: 0.9984 - val_loss: 1.1840 - val_acc: 0.8136
Epoch 463/500
557/557 [=====] - 3s 5ms/step - loss: 0.0347 - acc: 0.9984 - val_loss: 1.1794 - val_acc: 0.8136
Epoch 464/500
557/557 [=====] - 3s 6ms/step - loss: 0.0344 - acc: 0.9985 - val_loss: 1.1809 - val_acc: 0.8131
Epoch 465/500
557/557 [=====] - 3s 5ms/step - loss: 0.0343 - acc: 0.9987 - val_loss: 1.1826 - val_acc: 0.8137
Epoch 466/500
557/557 [=====] - 3s 5ms/step - loss: 0.0341 - acc: 0.9988 - val_loss: 1.1860 - val_acc: 0.8132
Epoch 467/500
557/557 [=====] - 3s 5ms/step - loss: 0.0340 - acc: 0.9987 - val_loss: 1.1856 - val_acc: 0.8131
Epoch 468/500
557/557 [=====] - 3s 5ms/step - loss: 0.0338 - acc: 0.9987 - val_loss: 1.1858 - val_acc: 0.8132
Epoch 469/500
557/557 [=====] - 3s 5ms/step - loss: 0.0337 - acc: 0.9987 - val_loss: 1.1855 - val_acc: 0.8134
Epoch 470/500
557/557 [=====] - 3s 5ms/step - loss: 0.0334 - acc: 0.9988 - val_loss: 1.1813 - val_acc: 0.8132
Epoch 471/500
557/557 [=====] - 3s 5ms/step - loss: 0.0333 - acc: 0.9988 - val_loss: 1.1850 - val_acc: 0.8123
Epoch 472/500
557/557 [=====] - 3s 5ms/step - loss: 0.0332 - acc: 0.9988 - val_loss: 1.1860 - val_acc: 0.8126
Epoch 473/500
557/557 [=====] - 3s 5ms/step - loss: 0.0330 - acc: 0.9988 - val_loss: 1.1752 - val_acc: 0.8134
Epoch 474/500
557/557 [=====] - 3s 5ms/step - loss: 0.0329 - acc: 0.9988 - val_loss: 1.1805 - val_acc: 0.8131
Epoch 475/500
557/557 [=====] - 3s 5ms/step - loss: 0.0327 - acc: 0.9988 - val_loss: 1.1797 - val_acc: 0.8121
Epoch 476/500
557/557 [=====] - 3s 6ms/step - loss: 0.0326 - acc: 0.9988 - val_loss: 1.1777 - val_acc: 0.8124
Epoch 477/500
557/557 [=====] - 3s 5ms/step - loss: 0.0324 - acc: 0.9988 - val_loss: 1.1746 - val_acc: 0.8131
Epoch 478/500
557/557 [=====] - 3s 5ms/step - loss: 0.0322 - acc: 0.9988 - val_loss: 1.1756 - val_acc: 0.8122
Epoch 479/500
557/557 [=====] - 3s 5ms/step - loss: 0.0321 - acc: 0.9988 - val_loss: 1.1745 - val_acc: 0.8124
Epoch 480/500
557/557 [=====] - 3s 5ms/step - loss: 0.0319 - acc: 0.9988 - val_loss: 1.1748 - val_acc: 0.8139
Epoch 481/500
557/557 [=====] - 3s 5ms/step - loss: 0.0318 - acc: 0.9988 - val_loss: 1.1759 - val_acc: 0.8142
Epoch 482/500
557/557 [=====] - 3s 5ms/step - loss: 0.0316 - acc: 0.9988 - val_loss: 1.1780 - val_acc: 0.8148
Epoch 483/500
557/557 [=====] - 3s 5ms/step - loss: 0.0315 - acc: 0.9988 - val_loss: 1.1808 - val_acc: 0.8148
Epoch 484/500
557/557 [=====] - 3s 6ms/step - loss: 0.0313 - acc: 0.9989 - val_loss: 1.1928 - val_acc: 0.8158
Epoch 485/500
557/557 [=====] - 3s 5ms/step - loss: 0.0312 - acc: 0.9990 - val_loss: 1.1969 - val_acc: 0.8158
Epoch 486/500
557/557 [=====] - 3s 5ms/step - loss: 0.0310 - acc: 0.9990 - val_loss: 1.1997 - val_acc: 0.8164
Epoch 487/500
557/557 [=====] - 3s 5ms/step - loss: 0.0308 - acc: 0.9990 - val_loss: 1.2020 - val_acc: 0.8158
Epoch 488/500
557/557 [=====] - 3s 5ms/step - loss: 0.0307 - acc: 0.9990 - val_loss: 1.2108 - val_acc: 0.8156
Epoch 489/500
557/557 [=====] - 3s 5ms/step - loss: 0.0306 - acc: 0.9990 - val_loss: 1.2180 - val_acc: 0.8154
Epoch 490/500
557/557 [=====] - 3s 5ms/step - loss: 0.0304 - acc: 0.9990 - val_loss: 1.2295 - val_acc: 0.8117
Epoch 491/500
557/557 [=====] - 3s 5ms/step - loss: 0.0303 - acc: 0.9990 - val_loss: 1.2326 - val_acc: 0.8110
Epoch 492/500
557/557 [=====] - 3s 6ms/step - loss: 0.0301 - acc: 0.9990 - val_loss: 1.2498 - val_acc: 0.8085
Epoch 493/500
557/557 [=====] - 3s 5ms/step - loss: 0.0300 - acc: 0.9990 - val_loss: 1.2511 - val_acc: 0.8089
Epoch 494/500
557/557 [=====] - 3s 5ms/step - loss: 0.0298 - acc: 0.9990 - val_loss: 1.2620 - val_acc: 0.8078
Epoch 495/500
557/557 [=====] - 2s 4ms/step - loss: 0.0297 - acc: 0.9990 - val_loss: 1.2677 - val_acc: 0.8063
Epoch 496/500
557/557 [=====] - 3s 5ms/step - loss: 0.0295 - acc: 0.9990 - val_loss: 1.2698 - val_acc: 0.8067
Epoch 497/500
557/557 [=====] - 3s 5ms/step - loss: 0.0294 - acc: 0.9990 - val_loss: 1.2781 - val_acc: 0.8057
Epoch 498/500
557/557 [=====] - 3s 5ms/step - loss: 0.0293 - acc: 0.9990 - val_loss: 1.2794 - val_acc: 0.8075
Epoch 499/500
557/557 [=====] - 3s 5ms/step - loss: 0.0291 - acc: 0.9990 - val_loss: 1.2793 - val_acc: 0.8082
Epoch 500/500
557/557 [=====] - 3s 5ms/step - loss: 0.0289 - acc: 0.9990 - val_loss: 1.2864 - val_acc: 0.8058
```

```
In [12]: wildflower_classifier.eda_summary()
```

/home/marlog/miniconda2/envs/keras-tf-gpu-marlog2/lib/python3.5/site-packages/bokeh/core/json_encoder.py:80: FutureWarning: Conversion of the second argument of issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 == np.dtype(float).type`.
elif np.issubdtype(type(obj), np.float):



```
In [13]: wildflower_classifier.evaluate_top_model_training()
```



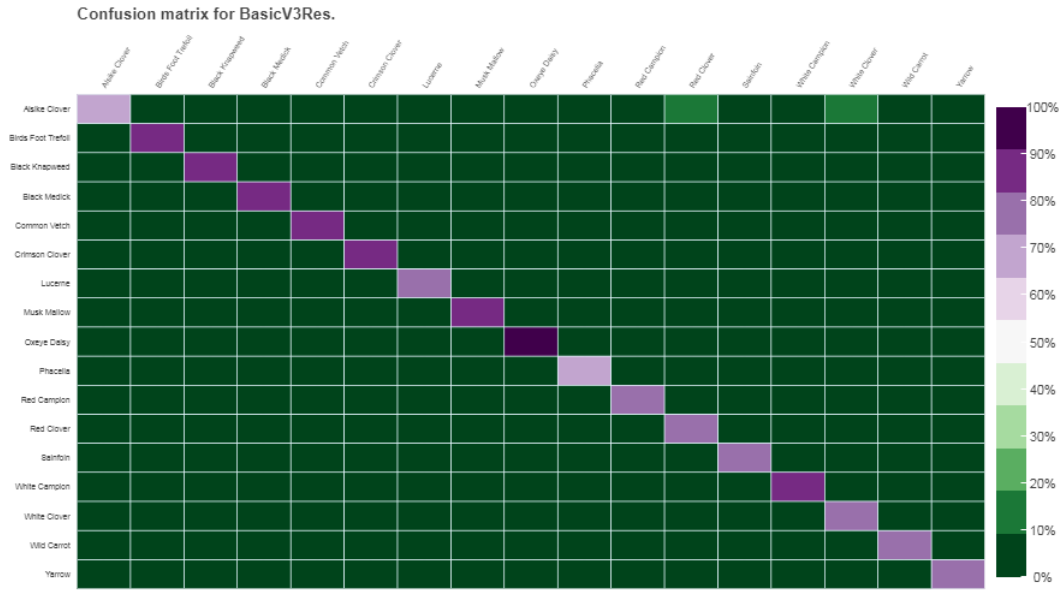
```
In [14]: wildflower_classifier.evaluate_model_prediction("accuracy")
```

Creating predictions using BasicV3Res for the validation data.
 Classification Accuracy for BasicV3Res is: 0.8267251560654632

```
In [15]: wildflower_classifier.evaluate_model_prediction("confusion")
```

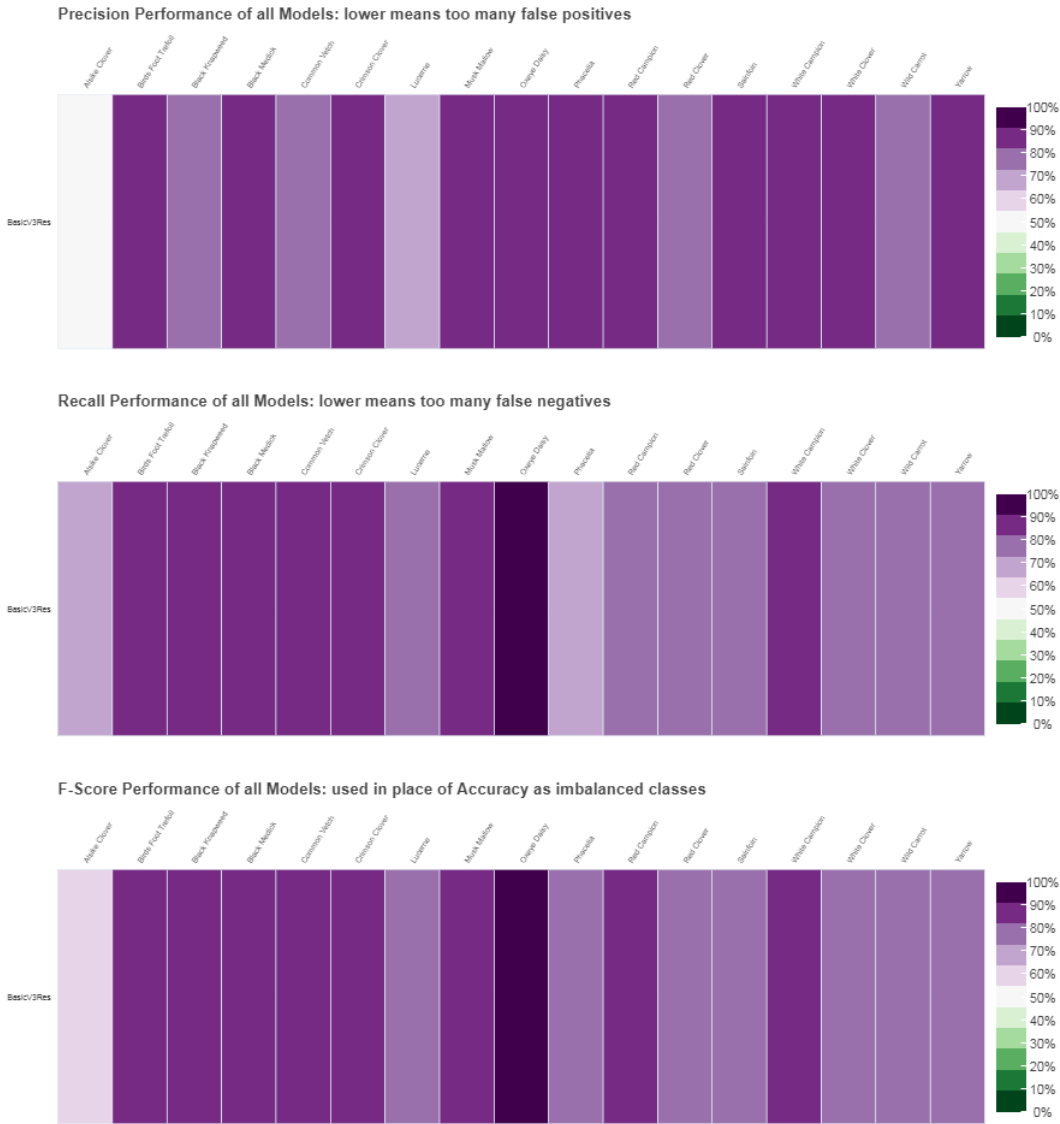
WARNING:root:Using previously created predictions for model: BasicV3Res.
WARNING:root:Using previously loaded predictions.

	Model	Class	normalised
0	Alsike Clover	Alsike Clover	69.747899
18	Birds Foot Trefoil	Birds Foot Trefoil	87.719298
36	Black Knapweed	Black Knapweed	90.476190
54	Black Medick	Black Medick	85.714286
72	Common Vetch	Common Vetch	83.919598
90	Crimson Clover	Crimson Clover	85.873606
108	Lucerne	Lucerne	73.858921
126	Musk Mallow	Musk Mallow	90.476190
144	Oxeye Daisy	Oxeye Daisy	96.992481
162	Phacelia	Phacelia	70.927318
180	Red Campion	Red Campion	81.453634
198	Red Clover	Red Clover	79.197995
216	Sainfoin	Sainfoin	76.691729
234	White Campion	White Campion	89.473684
252	White Clover	White Clover	73.433584
270	Wild Carrot	Wild Carrot	80.701754
288	Yarrow	Yarrow	76.441103




```
In [16]: wildflower_classifier.evaluate_model_prediction("heatmaps")
```

WARNING:root:Using previously created predictions for model: BasicV3Res.
WARNING:root:Using previously loaded predictions.



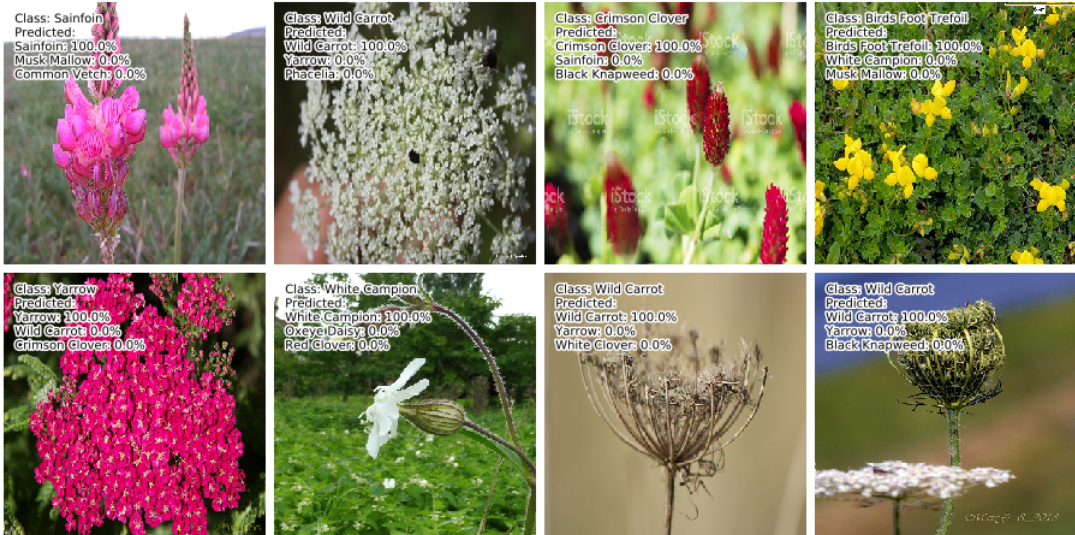
```
In [17]: wildflower_classifier.evaluate_model_prediction("extremes-examples")
```

WARNING:root:Using previously created predictions for model: BasicV3Res.
WARNING:root:Using previously loaded predictions.

The best performing images are calculated as those that have the correct predicted class, ordered by those with the smallest distance between the prediction for the correct class and a perfect prediction.

The worst performing images are calculated as those that have the wrong predicted class, ordered by those with the largest distance between the prediction for the correct class and a perfect prediction.
TODO: Maybe this should instead be how many classes are between the first prediction and the actual class prediction position?:

Best performing images for model: BasicV3Res

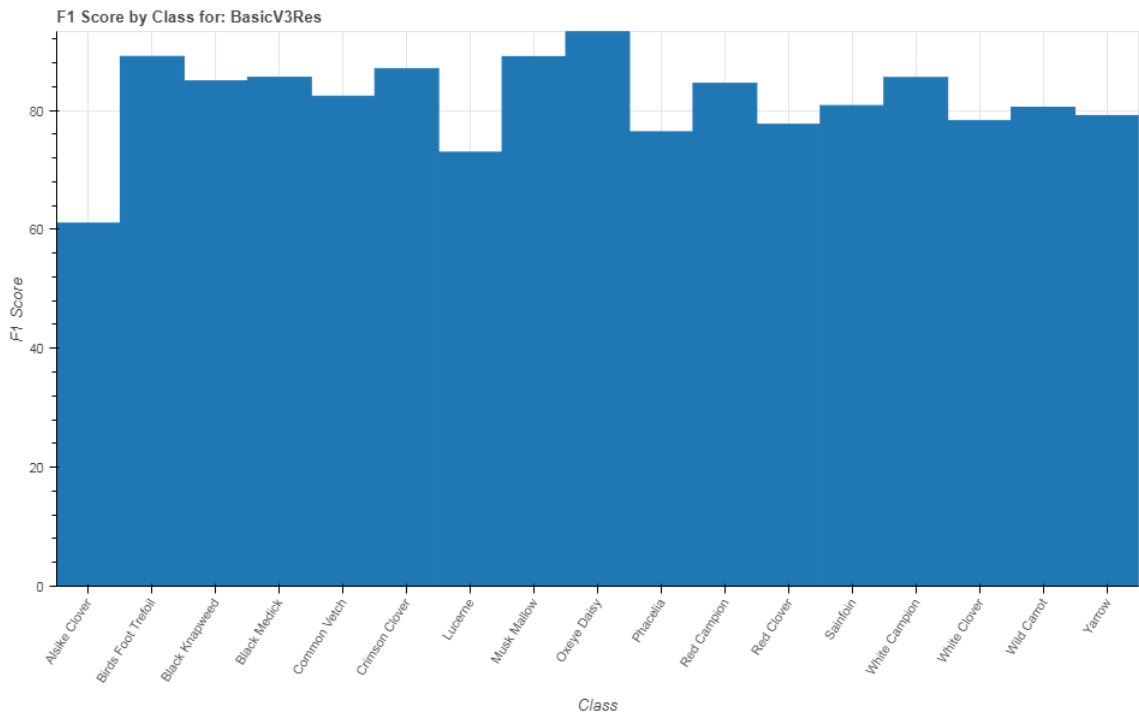


Worst performing images for model: BasicV3Res



```
In [18]: wildflower_classifier.evaluate_model_prediction("bar-f1score")
```

WARNING:root:Using previously created predictions for model: BasicV3Res.
WARNING:root:Using previously loaded predictions.



```
In [19]: wildflower_classifier.evaluate_model_prediction("line-f1score")
```

WARNING:root:Using previously created predictions for model: BasicV3Res.
WARNING:root:Using previously loaded predictions.

