Network Structure:

```
AlexNet(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(11, 11), stride=(4, 4),
         padding=(2, 2))
    (1): ReLU(inplace)
    (2): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1,
         ceil_mode=False)
    (3): Conv2d(64, 192, kernel_size=(5, 5), stride=(1, 1),
         padding=(2, 2))
    (4): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
         track_running_stats=True)
    (5): ReLU(inplace)
    (6): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1,
         ceil_mode=False)
    (7): Conv2d(192, 384, kernel_size=(3, 3), stride=(1, 1),
         padding=(1, 1))
    (8): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True,
         track_running_stats=True)
    (9): ReLU(inplace)
    (10): Conv2d(384, 256, kernel_size=(3, 3), stride=(1, 1),
          padding=(1, 1))
    (11): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
          track_running_stats=True)
    (12): ReLU(inplace)
    (13): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
          padding=(1, 1))
    (14): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
          track_running_stats=True)
    (15): ReLU(inplace)
    (16): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
          padding=(1, 1))
    (17): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
          track_running_stats=True)
    (18): ReLU(inplace)
    (19): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1,
          ceil_mode=False)
  )
  (classifier): Classifier(
    (layers): ModuleDict(
      (fc0): Linear(in_features=9216, out_features=1024, bias=True)
      (bn0): BatchNorm1d(1024, eps=1e-05, momentum=0.1, affine=True,
             track_running_stats=True)
      (fc1): Linear(in_features=1024, out_features=1024, bias=True)
      (bn1): BatchNorm1d(1024, eps=1e-05, momentum=0.1, affine=True,
             track_running_stats=True)
    )
    (affine): Linear(in_features=1024, out_features=14, bias=True)
  )
)
```

Feature Layers:

I have frozen the feature layers 0, 1 and 2 in the alexnet model. However, starting from feature 3 I have trained the model. Here is feature 3 according to the model sketched above:

```
3): Conv2d(64, 192, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
```

Moreover, I have inserted a 2D Batch Normalization before every ReLU starting from index 4 in the feature layers. Finally, I have added an additional Conv2d layer with 256 filters as feature layer 16:

```
(16): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
```

Classification Layers:

For the classifier layer I have used a fully connected layer with two hidden layers of size 1024 and I have used 1D Batch Normalization before every ReLU operation and after every hidden layer.

Training Process:

Data Augmentation:

- For every image I have created 4 crops with gaussian noise added to the crop coordinates (mean=0, std=3).
- For every cropped photo I have created a horizontal flip as well.
- For every cropped photo and its horizontal flip, I have altered the brightness with gaussian brightness factor (mean=1.5, std=0.3)
- All the augmented images are first resized to 120x120 due to RAM issues.
- The final images are resized to 224x224 in the __getitem__ method when they are retrieved by the data loader.

Learning Rate Parameters:

- I have used Adam as my optimizer.
- I used learning rate of 1e-4 for epochs 0-5
- I used learning rate of 1e-5 for epochs 5-10
- I used learning rate of 1e-6 for epochs 10-20
- I used learning rate of 1e-7 for epochs 20-25

Accuracy on Test Data:

The keys are the radius in pixel and the values are the accuracy on the test dataset.

```
{5.0: 76.62467079507438,
10.0: 98.42693430137376,
15.0: 99.83628728023348,
20.0: 99.97864616698698,
25.0: 99.97864616698698,
30.0: 99.98576411132464,
35.0: 99.99288205566232,
40.0: 100.0,
45.0: 100.0}
```

Final Notes:

- I have tested vgg_11_bn, inception_v3, resent18, resnet34 and this is the best result I was able to achieve.
- I had no collaborator for this project and I'm submitting this individually.