

- Type of keypoints:
SIFT
- Link to the off-the-shelf software for keypoint detection:
https://docs.opencv.org/3.4.3/da/df5/tutorial_py_sift_intro.html
- Hyper-parameters of CNN2 and CNN3:
 CNN2:


```
self.features = nn.Sequential(
    nn.Conv2d(1, 64, kernel_size=3, stride=2, padding=1, bias = False),
    nn.BatchNorm2d(64, affine=False),
    nn.ReLU(),
    nn.Conv2d(64, 128, kernel_size=3, stride=1, padding=1, bias = False),
    nn.BatchNorm2d(128, affine=False),
    nn.ReLU(),
    nn.Conv2d(128, 128, kernel_size=3, stride=2, padding=1, bias = False),
    nn.BatchNorm2d(128, affine=False),
    nn.ReLU(),
    nn.Conv2d(128, 128, kernel_size=3, stride=1, padding=1, bias = False),
    nn.BatchNorm2d(128, affine=False),
    nn.ReLU(),
    nn.Dropout(0.3),
    nn.Conv2d(128, 128, kernel_size=8, bias = False),
    nn.BatchNorm2d(128, affine=False),
)
```

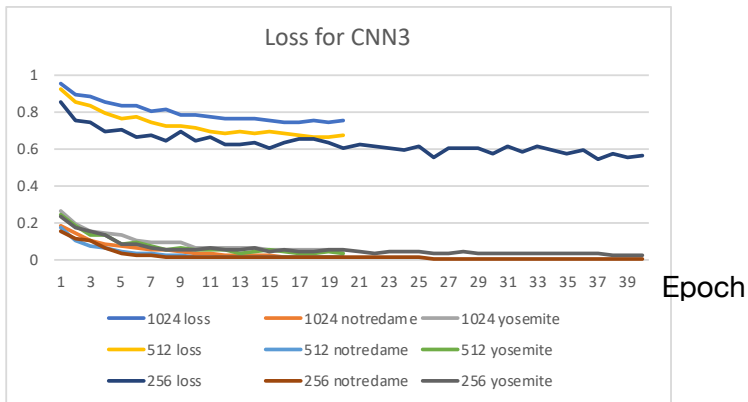
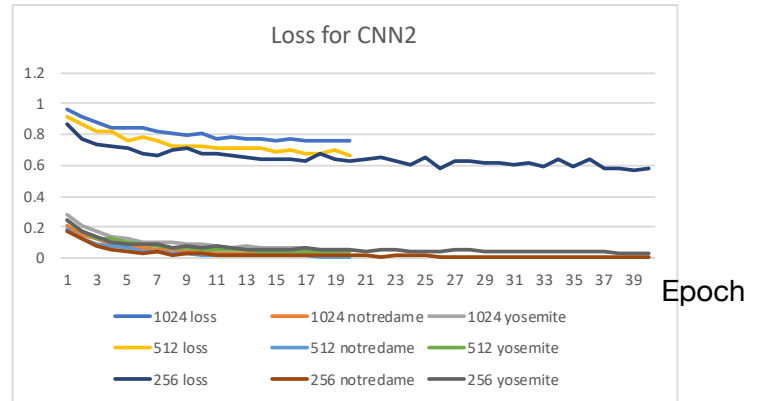
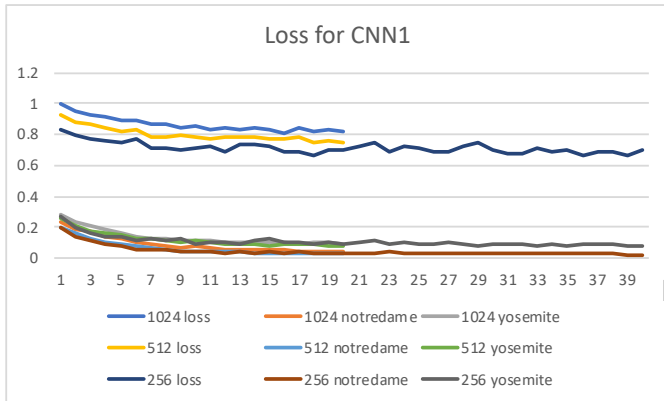
 CNN3:


```
self.features = nn.Sequential(
    nn.Conv2d(1, 64, kernel_size=3, stride=2, padding=1, bias = False),
    nn.BatchNorm2d(64, affine=False),
    nn.ReLU(),
    nn.Conv2d(64, 128, kernel_size=3, stride=1, padding=1, bias = False),
    nn.BatchNorm2d(128, affine=False),
    nn.ReLU(),
    nn.Conv2d(128, 128, kernel_size=3, stride=1, padding=1, bias = False),
    nn.BatchNorm2d(128, affine=False),
    nn.ReLU(),
    nn.Conv2d(128, 128, kernel_size=3, stride=1, padding=1, bias = False),
    nn.BatchNorm2d(128, affine=False),
    nn.ReLU(),
    nn.Conv2d(128, 128, kernel_size=3, stride=2, padding=1, bias = False),
    nn.BatchNorm2d(128, affine=False),
    nn.ReLU(),
    nn.Conv2d(128, 128, kernel_size=3, stride=1, padding=1, bias = False),
```

CS537 HW1

```
nn.BatchNorm2d(128, affine=False),
nn.ReLU(),
nn.Dropout(0.3),
nn.Conv2d(128, 128, kernel_size=8, bias = False),
nn.BatchNorm2d(128, affine=False),
)
```

- Plots of the training loss:



- Optimal mini-batch size and optimal epochs:

Optimal value	CNN1	CNN2	CNN3
mini-batch size	256	256	256
epoch	38	38	36
loss /notredame /yosemite)	0.665076 /0.02526 /0.0769	0.569781 /0.00798 /0.0362	0.547276 /0.00726 /0.03244

Due to the fact that the losses of some mini-batch sizes are still decreasing, if the mini-batch size is smaller and the epoch is larger, the loss may be further decreased.