

#### Task 4. CIFAR-10 - 1-NN classifier (20 points)

Here are some stats from my implementation:

Distance measure used: Absolute difference

CPU used	Accuracy	Execution time	Threads	Picture
Intel i5-3230M (4) @ 3.200GHz	38.59%	3503s $\approx$ 1h	4	1
Intel Core i5-8400 @ 2.80GHz	38.59%	2158s $\approx$ 36 min	4	2
Intel Core i5-8400 @ 2.80GHz	38.59%	2271s $\approx$ 38 min	8	3

```
85     pool = Pool(n)
86     start = time.time()
87     predictedLabels = pool.map(predict_labels, f
Line 87, Column 89

[tuomas@archlinux Exercise 2]$ python script7.py
training set size: (50000, 3072)
testing set size: (10000, 3072)
Starting search...
Starting search...
Starting search...
Starting search...
10000 10000
Execution time: 3503.2580156326294 s
1NN classifier accuracy: 38.59 %
[tuomas@archlinux Exercise 2]$
```

Picture 1.

```
C:\Tuomas_Python>python task4.py
training set size: (50000, 3072)
testing set size: (10000, 3072)
training set size: (50000, 3072)
training set size: (50000, 3072)
training set size: (50000, 3072)
Starting search...
Starting search...
Starting search...
Starting search...
Execution time: 2157.8958356380463 s
1NN classifier accuracy: 38.59 %

79     testX, _ = data_partition(testX,
80     # Create threadpool with 4 threa
81     pool = Pool(4)
82     start = time.time()
83     predictedLabels = pool.map(predi
84
85     accuracy=class_acc(predictedLabe
86     print("Execution time:", time.ti
87     print("1NN classifier accuracy:"
88
89     # Define these as a global variables.
90     trainX, trainY = load_whole_trainset
91     if name == 'main':
```

Picture 2.

```

C:\Tuomas_Python>python task4.py
training set size: (50000, 3072)
testing set size: (10000, 3072)
training set size: (50000, 3072)
Starting search...
training set size: (50000, 3072)
training set size: (50000, 3072)
Starting search...
training set size: (50000, 3072)
Starting search...
training set size: (50000, 3072)
Starting search...
training set size: (50000, 3072)
Starting search...
Starting search...
Starting search...
Execution time: 2271.473070383072 s
1NN classifier accuracy: 38.59 %

```

```

75 #trainX2, trainY2 =
76 #trainY3 = np.vstack
77 # Load test data:
78 testX, testY = load
79 testX, _ = data_par
80 # Create threadpool
81 pool = Pool(8)
82 start = time.time()
83 predictedLabels = p
84 testX[7]])
85 predictedLabels=pred
86 predictedLabels[5]+p
87
88 accuracy=class_acc(p
89 print("Execution tim
90 print("1NN classifie
91
92 # Define these as a globa
93 trainX, trainY = load_wh

```

Picture 3.