1. E.)

How to rescale images down to a 32 X 32 X 3 color thumbnail image:

thumbnail = imresize(image, [32 32]);

Which of the 10 classifications do unknown objects get placed in?

Image 1:Tree in Field

Image 2: Wintery Mountain Scene

Image 3: Picture of Ryan and Zach in the Cybertorium

Image 4: Picture of Steve

Image 5: Picture of Sean

Image 6: Picture of Desktop Computer

Image 7: Picture of Football on Field

Image 8: Picture of Robert Collins Standing in Front of a White-Board

Image 9: Picture of Old Main

Image 10: Picture of Penn State Logo

|  |  |
| --- | --- |
|  | Classification |
| Image 1 | Horse (8) |
| Image 2 | Ship (9) |
| Image 3 | Bird (3) |
| Image 4 | Frog (7) |
| Image 5 | Cat (4) |
| Image 6 | Automobile (2) |
| Image 7 | Deer (5) |
| Image 8 | Truck (10) |
| Image 9 | Bird (3) |
| Image 10 | Truck (10) |

How does well does the CNN work on 110 images pulled from the web?

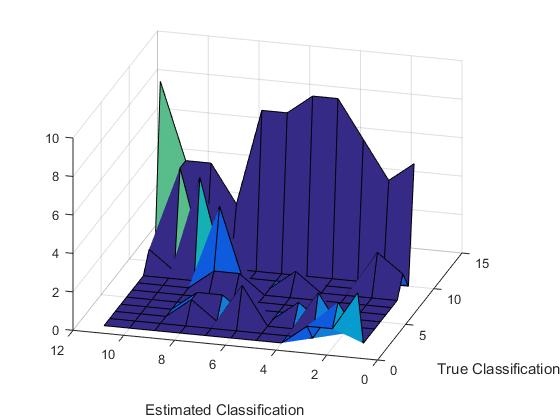
We got 10 images from the web for each of the 10 classifications and the 10 images of unknown classification described above. In order to determine if an image fell into the unknown classification we took the difference between the two max probabilities and if the difference was less than 0.25 then we classified the image as unknown. We choose the metric because we figured that if an image is unknown the CNN should pick multiple classifications for it to land in.

Here are our results:

The classification numbers 1-10 are the same classification number 11 is the unknown classification.

Here is our confusion matrix:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | CNN  Estimated  Classification | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| True  Classification |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 |  | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 6 |
| 2 |  | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 5 |
| 3 |  | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7 |
| 4 |  | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 9 |
| 5 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 9 |
| 6 |  | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 |
| 7 |  | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 8 |
| 8 |  | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 4 | 0 | 0 | 3 |
| 9 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 5 |
| 10 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 5 |
| 11 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 9 |



Accuracy = 29.09%

As you can see this metric did a great job of identifying unknown pictures, but it also classified a lot of pictures as unknown that did not belong in the unknown category. So this clearly is not a very good metric for distinguishing between unknown classifications and the other classifications. We also tried looking at how many classifications had a probability above the mean for several pictures and that didn’t produce very good results either. We also tried looking at the derivative of the probability vector and that worked very poorly as well. In fact, that metric produce 0 mean Gaussian noise with a very low standard deviation.