Srikanth Kilaru

Question1:

I was able to generate data that had a training accuracy of
94.4% for J48 and 34.5% for IBk, which is about 60% difference.

The data is Q1.csv, and the code to generate this data is in a function
called problem1() in ps4.py

I generated a 1000 samples of 10 attributes each where each attribute was a
random number drawn from different distributions.
I fixed the first such sample as the "master" sample and I calculated the
Eucledian distance between the rest of the 999 samples and this master
sample. I broke down he distance into different bins spaced by 100 units
and gave each bin a different class name.

J48 does much better because it can find this binning very easily, whereas
IBk (1 NN) struggles with this as during 10 fold training, the one master
sample randomly gets left out from the training set and the NN calculations
are now going to find nearest neighbors that may not be from the same class
at a much higher rate and therefore the original class assignment is lost,
thereby coming up with a very poor training accuracy

Question2:

I was able to generate data that had a training accuracy of
89.3% for MLP and 33.9% for NB, which is about 56% difference.

The data is Q2.csv, and the code to generate this data is in a function
called problem2() in ps4.py

I generated a 1000 samples of 10 attributes each where each attribute was a
random number drawn from the same distribution.
I calculated a weighted sum of these 10 attributes where the weights were
arbitrarily fixed by me.
sample. I broke down this weighted sum into different bins spaced by 25 units
and gave each bin a different class name.

MLP does very well here because it is very well suited to find the
underlying linear function, whereas NB in a nutshell tries to estimate the
class based upon the frequencies of values of the attributes which were
randomly generated and moreover to achieve a certain sum (which is what
drives the classification binning) can be achieved in many different ways and
therefore will the NB classification will suffer.

Question 3

		DATA		
Lottery	**Winner**	**Lunch**	**Noon**	**Spam**
1	1	0	1	0
1	1	1	1	1
0	0	1	0	0

FREQUENCY TABLE

			Spam	Ham		Row Sums
P(Lottery)	0.6	**Lottery**	1	1		2
P(Winner)	0.6	**Winner**	1	1		2
P(Lunch)	0.6	**Lunch**	1	1		2
P(Noon)	0.6	**Noon**	1	1		2
		Column Sums	4	4		
		P(Spam)	0.4	0.6	**P(Ham)**	
		P(Lottery | Spam)	0.666666667	0.5	***P(Lottery | Ham)***	
		P(Winner | Spam)	0.666666667	0.5	***P(Winner | Ham)***	
		P(Lunch | Spam)	0.666666667	0.5	***P(Lunch | Ham)***	
		P(Noon | Spam)	0.666666667	0.5	***P(Noon | Ham)***	

			Winner	**Accurate**
Sample 1	0.039506173	0.0375	Spam	No
Sample2	0.079012346	0.0375	Spam	Yes
Sample3	0.019753086	0.0375	Ham	Yes

With this dataset we can see that NB has a 2/3 training accuracy.
I believe this is because the attributes Winner and Noon are not really independent (they are both either 1 or 0), whereas NB assumes independence of all the attributes.

Whereas Logistic Regression can fit the data perfectly using Gradient ascent and therefore can find the ideal weights to fit the training dataset achieving 100% training accuracy.

Question 4

(a)
Number of parameters =
Num(HoursSleep)*Num(Studied)*Num(LikesMaterial)*Num(ExamScore) - 1
= 9*3*3*3 – 1
= 242

(b)
P(ExamScore | HoursSleep, Studied, LikesMaterial)
= (3-1)*9*3*3

= 162

(c)
Assuming independence:
P(ExamScore | HoursSleep, Studied, LikesMaterial) =
P(HoursSleep | ExamScore) * P(Studied | ExamScore) * P(LikesMaterial | ExamScore) *
P(ExamScore)
= (9-1)*3 + (3-1)*3 + (3-1)*3 + (3-1)
= 24 + 6 + 6 + 2
= 38

Question 5

It is in a function called cosine_similarity() in ps4.py

Question 6

similarity between MJ1 and MJ2 using pixel representation is 0.3708619464542405
similarity between MJ1 and Cat using pixel representation is 0.4730885235789724
similarity between MJ2 and Cat using pixel representation is 0.6197439905980221
similarity between MJ1 and MJ2 using VGG representation is 0.9600110261784119
similarity between MJ1 and Cat using VGG representation is 0.15253511266679187
similarity between MJ2 and Cat using VGG representation is 0.14183421213603353

The most similarity using Pixel is between: MJ2 and Cat
The most similarity using VGG is between: MJ1 and MJ2

Question 7

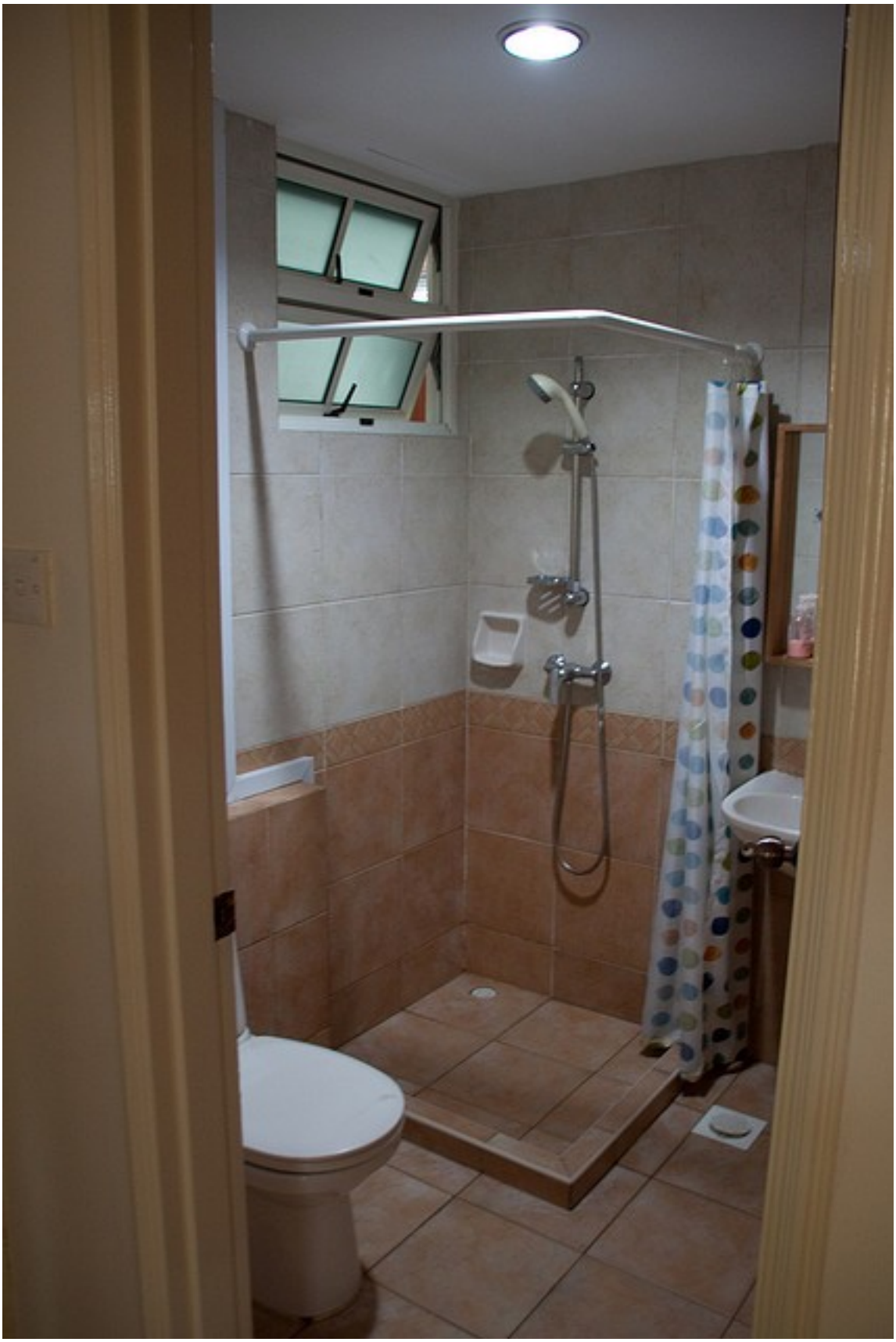
The problem with pixel representation is that it is a raw representation of the RGB intensities per pixel whereas the VGG representation is a result of several layers of convolution and pooling which is the way to extract the spatial structure and features of an image. To find similarities between two images we need information of the features like Histogram of Gradients (HoG) and edges etc. which are provided only by layers within a CNN like VGG.

Question 8

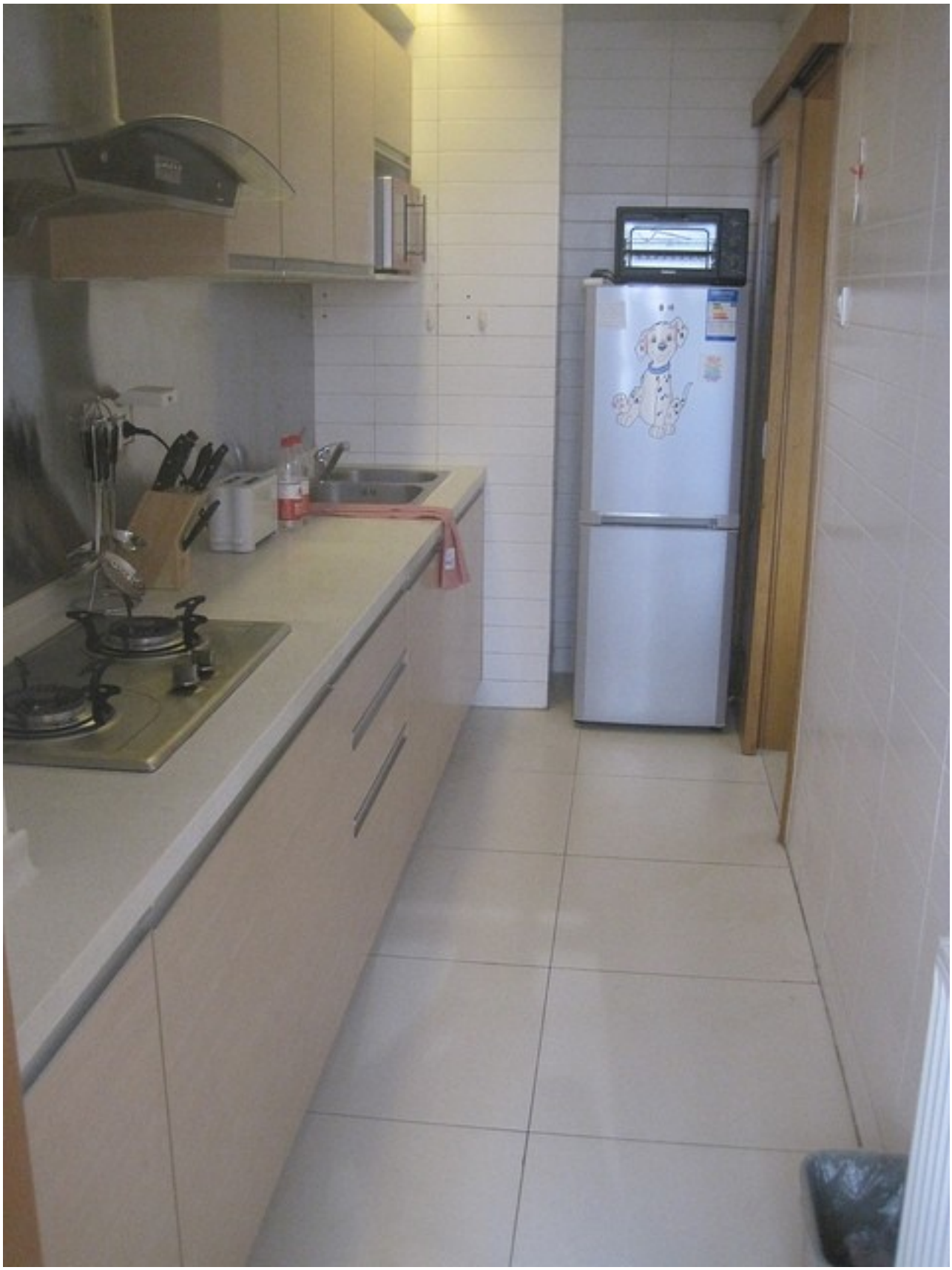
Yes, the VGG captioning scheme is much better than the pixel based scheme in finding image similarity and captioning. As mentioned in the answer for Question 7, VGG scheme finds the right image feature level similarity and therefore is able to inherit the right caption from the train image.

Question 9

For VGG:
A caption that it does not get right is -
Picture of bathroom with shower in far corner and toilet sitting nearest to the door.



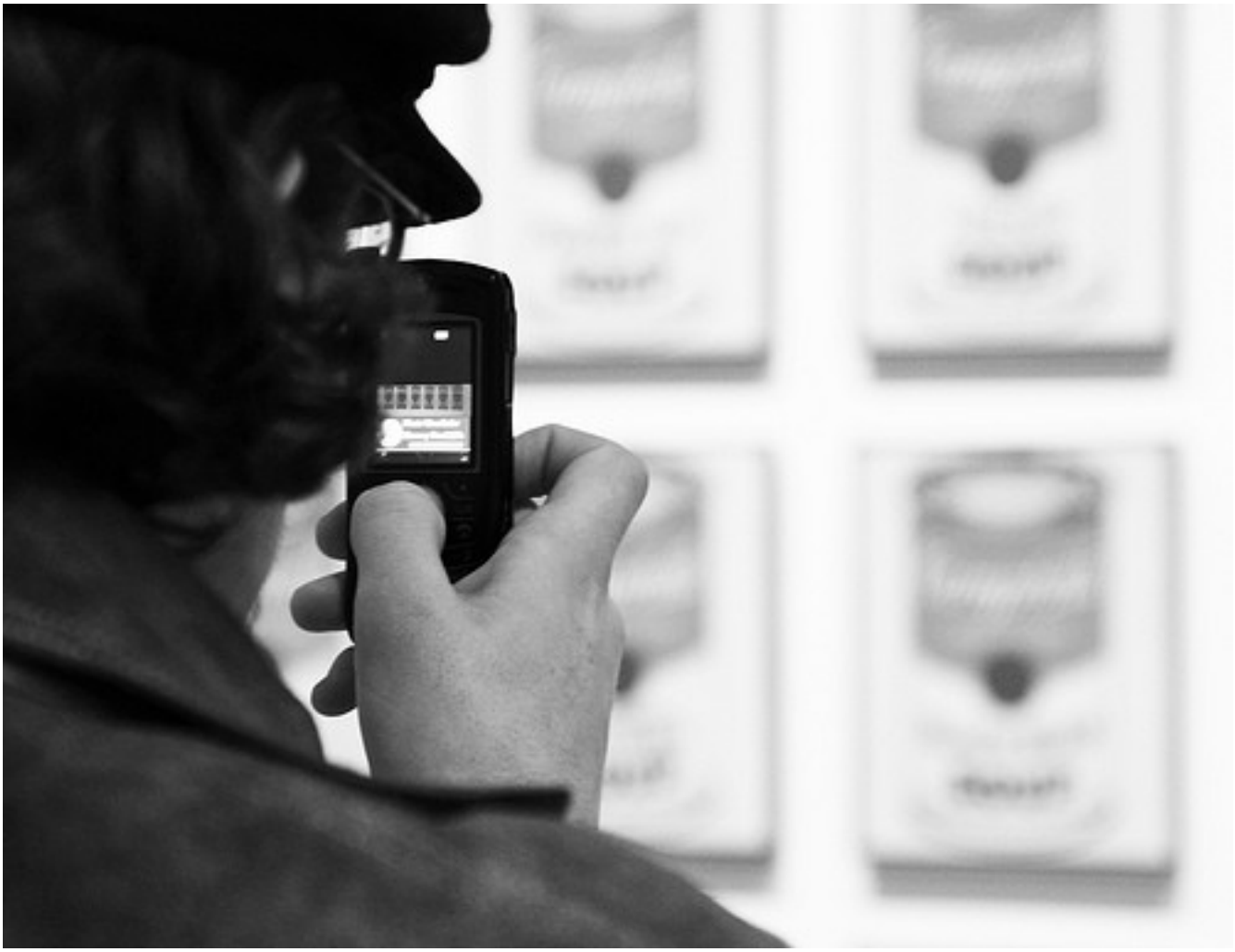
Correct captioning in training



Incorrect captioning using VGG
In my opinion, the reason why VGG mis-captions this image is because it finds similarities between the tall vertical edges thatare common between these two images and also the depth that is represented in both. In addition the tall shower curtain and the tall fridge on the far end look similar to mention a few similarities.

For Pixel, a caption that I found inappropriate is -
Darth Vader in a bathroom holding a toothbrush in one hand and staring at it.

Training image with the right captioning



Test image with mis-captioning

I believe that the reason these two images were found to be similar in the pixel scheme is because they are both grayscale and have the same segments/areas of the image with similar black portions and white portions. In other words from a gray-scale and location of these black/white/gray pixels in the image vector these two images are very similar and therefore they were picked at nearest neighbors.