RDFIA - Homework 3

Johan Pardo - Shubhamkumar Patel

January 14, 2022

1 Work 3-a

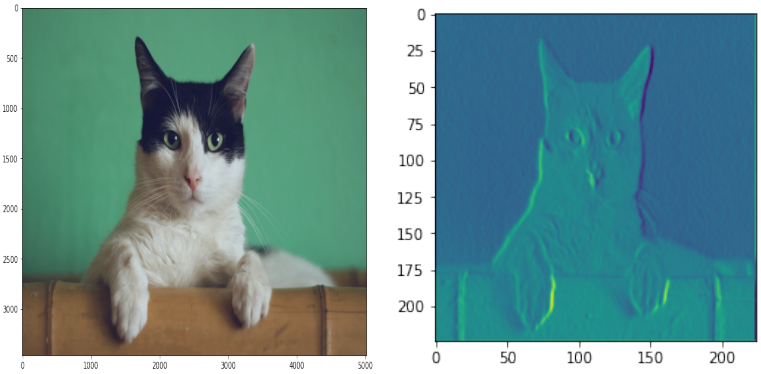
1.1 Section 1: VGG16 Architecture

Question 1: Knowing that the fully-connected layers account for the majority of the parameters in a model, give an estimate on the number of parameters of VGG16 (using the sizes given in Figure 1).
Knowing the fully-connected layers account for the majority of parameters in a model an estimation on the number of parameters of VGG16 will be around 123 millions parameters

Question 2: What is the output size of the last layer of VGG16? What does it correspond to?
The output size of the last layer of VGG16 is of size 1000. This corresponds to the number of class the VGG16 is trained on.

Question 3: Apply the network on several images of your choice and comment on the results.
Le network works well enough if the image we present to it include classes that it has learned. But as we try different variety of classes it starts to be overwhelmed and yields incorrect results.

Question 4: Visualize several activation maps obtained after the first convolutional layer. How can we interpret them?
The first convolutional layers reveal low level information such as the details we can extract by using a sobel filter for contours.



1.2 Section 2: Transfer learning with VGG16 on 15 scene

Question 5: Why not directly train VGG16 on 15 Scene?
We cannot directly train VGG16 on 15Scene because there is a high risk of overfitting. VGG16 is a complex network with plently of parameters and it was made to be trained over ImageNet dataset containning over 14 million images. Whereas 15Scene merely has a few thousands so even by using multiple data augmentation methods we won't be able to overcome the lack of images and new data for the training process.

Question 6: How can pre-training on ImageNet help classification for 15 Scene?
Pre-training on ImageNet helps us in our classification for 15 scenes because we can expect that the first layer will behave the same way, as feature extraction layers, in the ImageNet's dataset compared to our own and therefore cut training time while still having the overall good performance of the VGG network.

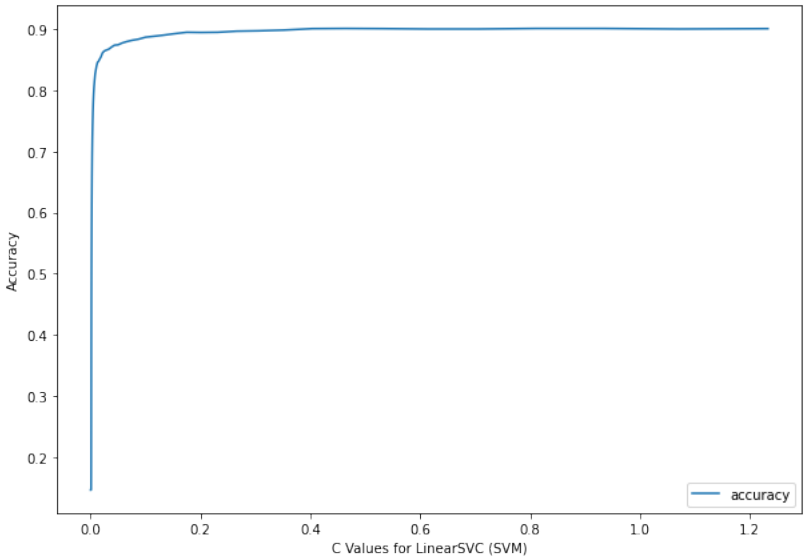
Question 7: What limits can you see with feature extraction?
The VGG16 being trained on one particular domain with one type of images, it might lack the ability to extract features from new images (for instance : black and white images). So for the feature extraction to make sense we require a domain that is as similar as the one used to train VGG in order for the feature extraction to work efficiently.

Question 8: What is the impact of the layer at which the features are extracted?
The layer at which the features are extracted determines the level of abstraction of the extraction. If we take the first layers we will keep abstract and low level information such as contours, forms, colors and shape but if we go deeper and closer to the output layer we will have high-level information such as the semantic information on objects or classes.

Question 9: The images from 15 Scene are black and white, but VGG16 requires RGB images. How can we get around this problem?
We can get around this problem by stacking the single channel (B and W image) into (RGB image) 3 channels by duplication. This way we get a RGB image that we can use as input for VGG16.

Question 10: Rather than training an independent classifier, is it possible to just use the neural network? Explain.
Yes, rather than training an independent classifier we can replace it by appending one fully connected layer to our VGG16 network. We can have 15 outputs for the 15Scene dataset classes using a Softmax activation.

Question 11: For every improvement that you test, explain your reasoning and comment on the obtained results.

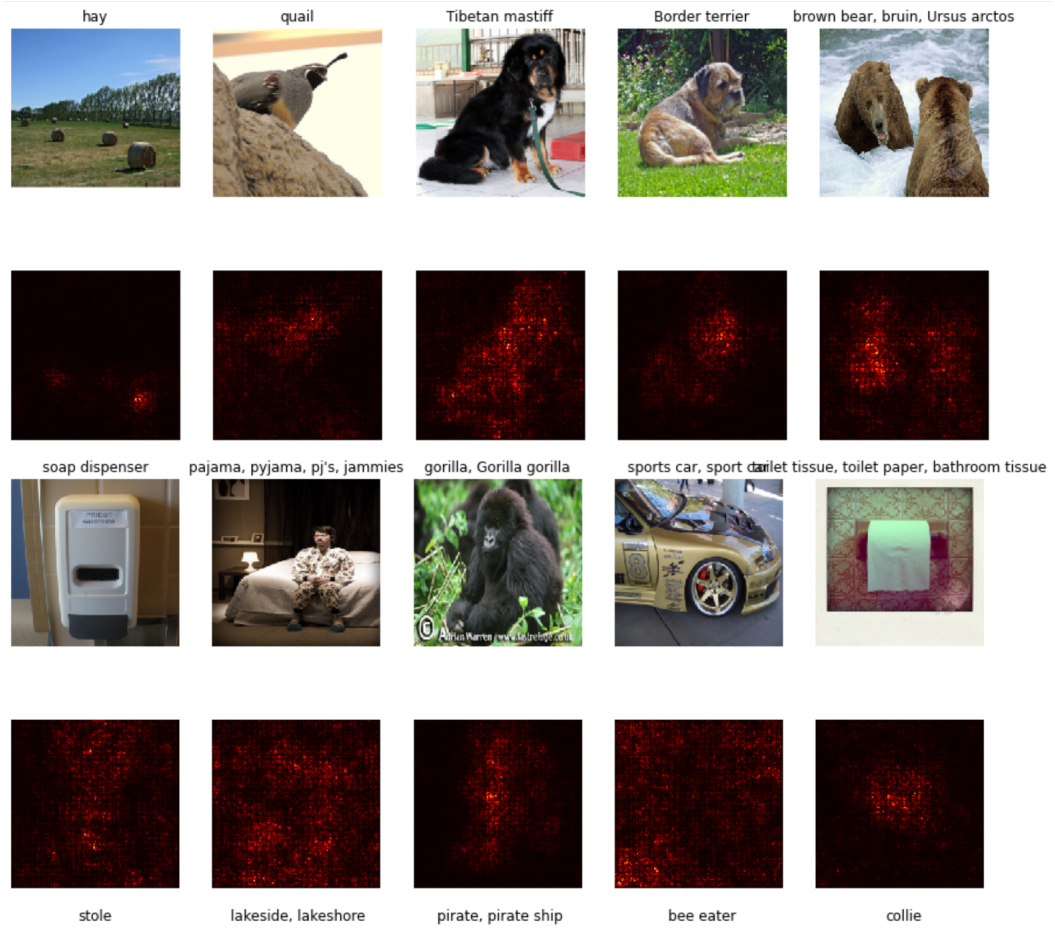


The C parameter being a hyperparameter of the LinearSVC (SVM) we tried to optimized it by trying different values. In the plot above we can see the evolution of the accuracy as we change the values of C. The best values of C is around C=1. Using an SVM limits out ability to fine-tune the performance of the model we can simply replace it with a Linear layear at the end of the network.

2 Work 3-b

2.1 Section 1 - Saliency Map

Question 1: Show and interpret the obtained result



We can see the points of interest in each images that show us that the saliency maps are activated around key points that describe a given class (dog, car, bird, etc...).

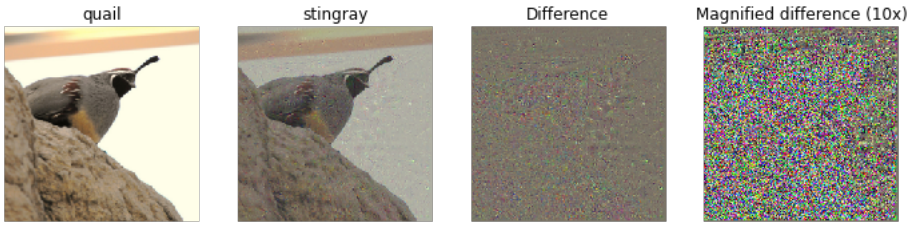
Question 2: Discuss the limits of this technique of visualizing the impact of different pixels.
The limit of this technique is that it is not linear and that we don't work on the whole image. This methods don't cover the whole image or objects of smaller sizes in the images. It yields noisy results.

Question 3: Can this technique be used for a different purpose than interpreting the network?
Yes this model can also be used for detection and segmentation but also to label classes in the image such as for instance based detection/segmentation methods.

Question 4: Test with a different network, for example VGG16, and comment
Using any other convolutional network we are going to end up getting similar results as the features extraction is performed in a similar way. Only the way the network perceives the features (more or less noise, variance, etc..) may vary but overall we get similar results.

2.2 Section 2 - Adversarial Examples

Question 5: Show and interpret the obtained

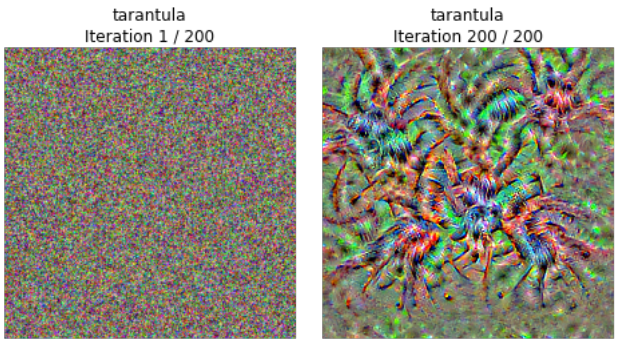


Using a base image from class 'quail' we were able to apply modifications that fooled the network into believing there was a 'stingray' class in the image, even if for human eyes it still looks as a 'quail'. We can see in the magnified difference image that we only introduced some noise which altogether was able to alter the prediction of the network.

Question 6: In practice, what consequences can this method have using convolutional neural network
We saw before that the network wasn't able to correctly predict the class quail, even if a human would not differentiate between the two images. This shows that the convolutional neural network are not reliable or accurate for all cases.

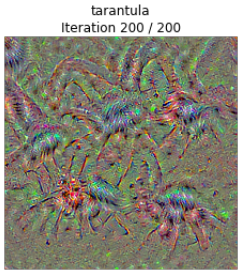
2.3 Section 3 - Class Visualization

Question 8: Show and interpret the obtained results.

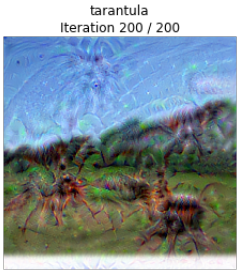


Beginning from a random noise the model tries to iteratively impose the class 'tarantula'. This lets us see what kind of information are relevant to the network and in our case the network's perception of curved lines with spike on it represents the class 'tarantula'.

Question 9: Try to vary the number of iterations and the learning rate as well as the regularization weight.

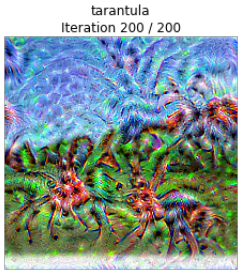


(a) $reg = 1, lr = 5$

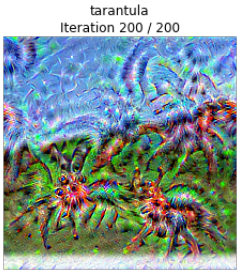


(b) $reg = 10^{-3}, lr = 1$

(c) $reg = 10^{-3}, lr = 10$



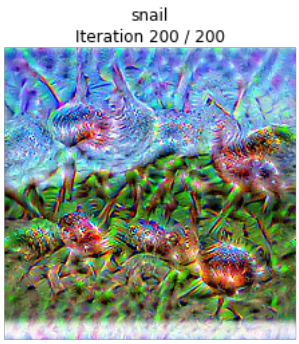
(d) $reg = 10^{-3}, lr = 5$



(e) $reg = 10^{-6}, lr = 5$

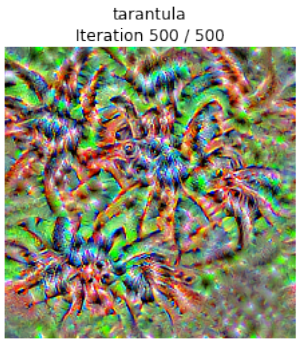
The regularization let's us emphasize a given class but we will lose information from the base image. And the learning rate is related to the weight of the model class, the higher it gets the more amplified the class's appearance gets.

Question 10: Try to use an image from ImageNet as the source image instead of a random image (parameter init_img). You can use the real class as the target class. Comment on the interest of doing this.

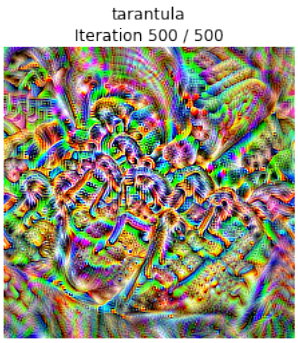


We start from a base image from ImageNet dataset and we do the same operations as before by replacing the random images by the Imagenet one. That we try to impose a class (snail) on the base image. What we see is the characteristic of class snail in the base image are amplified over the iterations.

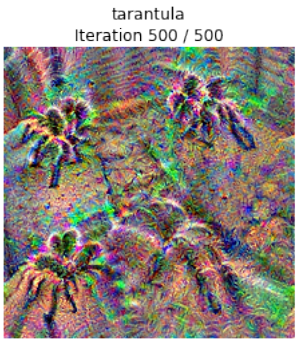
Question 11: Bonus: Test with another network, VGG16, for example, and comment on the results.



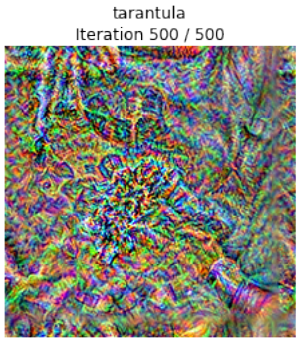
(a) SqueezeNet



(b) VGG16



(c) ResNet50



(d) InceptionV3

We can see that according to the network we select, they all have their own understanding or representation of the class "Tarantula" (Big/Small legs, More or less hairy legs, size of the tarantula, etc...).

3 Work 3-c

Question 1: If you keep the network with the three parts (green, blue, pink) but didn't use the GRL, what would happen ?
The goal of the GRL is to have a domain adaptation if we didn't have this part the model future layers will be harder to train. Without the GRL we won't be able to bring the domain loss higher which is required for the training process.

Question 2: Why does the performance on the source dataset may degrade a bit?
Because we have an domain adaptation, like the PCA, the GRL will keep only parts of the image that is necessary and therefore will loose information. By performing a domain adaptation we have altered the previous model by enforcing the characteristic of the new domain therefore because of this alteration the performance on the previous domain will be changed.

Question 3: Discuss the influence of the value of the negative number used to reverse the gradient in the GRL.
The goal of the GRL is to impose an increasing domain loss and have a decreasing class loss alongside. Without the negative numbers used to reverse the gradient we won't be able to have this effects.

Question 4: Another common method in domain adaptation is pseudo-labeling. Investigate what it is and describe it in your own words.
Pseudo labelling is the process of using the labelled data model to predict labels for unlabelled data. Here at first, a model has trained with the dataset containing labels and that model is used to generate pseudo labels for the unlabelled dataset. Finally, both the datasets and labels are combined for a final model training. It is called pseudo as these may or may not be real labels and we are generating them based on a similar data model.

4 Work 3-d

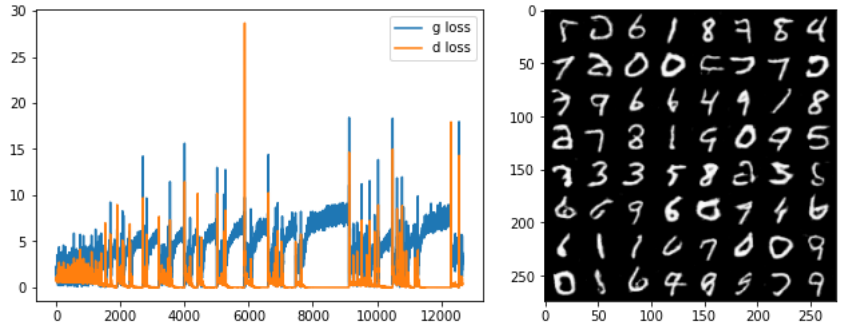
Question 1: Interpret the equations (6) and (7). What would happen if we only used one of the two ?
The equation number 6 is let us optimise the generator by maximizing the probability of D to predict true images. The equation number 7 optimises the discriminator by maximising the probability to well classify the real images and to discriminate the generated images by G. And therefore equation 5 is the right one that balances both equation.

Question 2: Ideally, what should the generator G transform the distribution P(z) to ?
At the end of the training the generator will have learned to transform distribution $P(z)$ to $Data$. By doing so it has learned to approximate real data from our dataset.

Question 3: Remark that the equation (6) is not directly derived from the equation 5. This is justified by the authors to obtain more stable training and avoid the saturation of gradients. What should the "true"equation be here ?
The "true" equation would have been :

$$min_g \mathbb{E}_{z=P(z)} log(1 - D(G(z)))$$

Question 4: Comment on the training of of the GAN with the default settings (progress of the generations, theloss, stability, image diversity, etc.



Looking at the generated images by the model we can say that it has managed to generate some pretty good and interpretable numbers just as we required from it. At the beginning the g and d loss tends to be close but at iteration pass one take the advantage over the other. In our case we consider that the "bottlenecks" come from the generator as it isn't able to produce better images for there to be a clear convergence.

Question 5: Comment on the diverse experiences that you have performed with the suggestions above. In particular,comment on the stability on training, the losses, the diversity of generated images, etc.

Here are a few experiments we did:

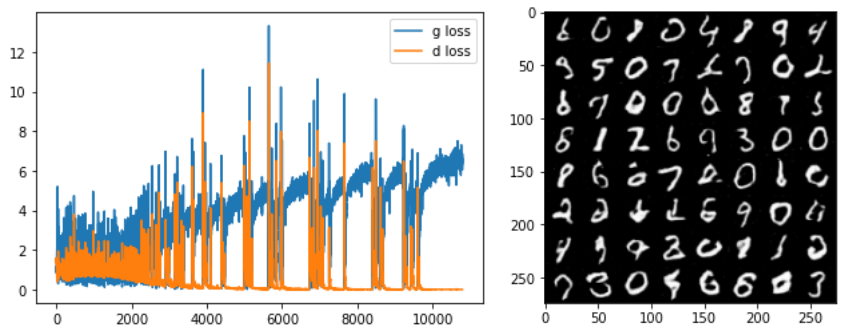


Figure 3: β=0.1

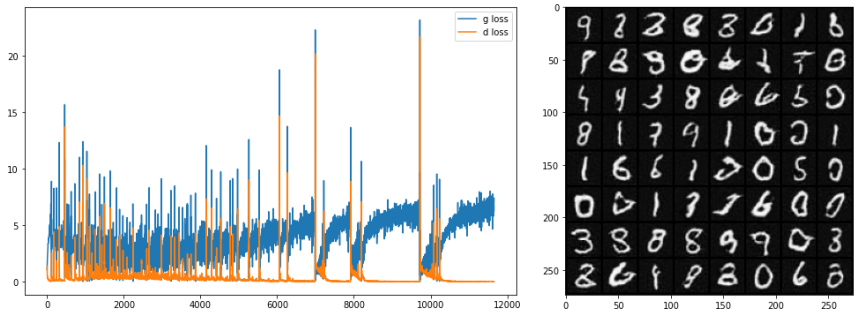


Figure 4: lr_d = lr_g = 0.00002

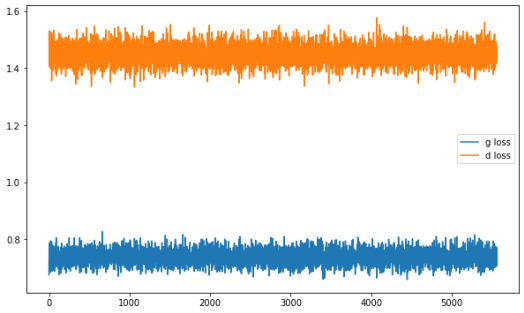


Figure 5: β=1

- $lr\_d = lr\_g = 0.0002$ In the basic GAN the lr_g was higher than the lr_d which I guessed lead to the big difference in in the difference of loss
- Beta is the decay term in the Adam optimizer we can see the effects of this term as smaller steps in the training and therefore better results in the end

Question 6: Comment on your experiences with the conditional DCGAN.
Compared to the GAN the DCGAN is harder to train we believe it comes from the fact that it has more parameters. We have experienced worsening quality in the generated images and the training losses are bad compared to what we had with GAN.

Question 7: Could we remove the vector y from the input of the discriminator (so having cD(x) instead of cD(x,y)) ?
If we remove the vector y from the input discriminator we would have a simple GAN that is not conditioned by anything so we would generate images that are not what we need. But in our case we don't want an unconditional GAN so we can't remove the vector y. If we don't keep y and we choose to have the class "1" as condition we can't skip the y parameter as the generator will continue to generate random images of classes other than the class of interest "1" without being penalised by the discriminator which itself won't know the condition that it should apply upon the generated images.

Question 8: Was your training more or less successful than the unconditional case ? Why ?
Compared to the cDCGAN, the cGAN is harder to train in order to get realistic images. During the training we get noisy images and the loss is quite high. SO compared to the unconditional case we are getting better results, the addition of the conditionning paramter helps to easily discriminate and penalise the generated images, hence the overall improvement we observed compared to the unconditional case.