McGill University
Faculty of Electrical, Computer and Software Engineering

HONORS THESIS
ECSE 499

IMAGE CAPTIONING

Sharhad Bashar
I.D: 206519664
Group: HT09

Supervisor of the Honors Thesis: Dr. Jeremy Cooperstock
Graduate Student: Roger Girgis
Study programme: Honors Electrical Engineering

April 11^{th}, 2017

Abstract

The purpose of this project is to build an image recognition system for Autour that will vocally identify and describe objects around the user. Autour is a eyes free mobile system that utilizes the camera, GPS and other built in sensors of a smartphone to give the visually impaired users a better sense of their surroundings. The built in camera will take pictures at certain time intervals and upload them to a computer via cloud. The computer will analyze and identify the different objects present in the image and describe them. Deep Neural Networks and additional image analysis will be utilized to identify the different objects and text present in the image. The caption generated will be sent back to the phone, and read out loud to the user. This project is aimed at the visually impaired community, to help them achieve a higher level of freedom and give them the best alternative to vision. During the length of the project, different image analysis and deep learning methods are to be tested and compared based on efficiency and performance.

Acknowledgment

Roger Girgis : Graduate Electrical Engineering Student | McGill University

Roger was a crucial member of this project, especially in mentoring and guiding me in the right directions. I was in constant contact with Roger throughout the semester, and we would meet to discuss different papers, approaches, and to get his feedback. He would keep me informed with relevant papers, information regarding the project and suggest teaching materials imperative to the thesis. He also advised on different tasks and relevant steps to keep moving forward. It was a great learning experience and pleasure to work with Mr. Girgis.

Shovan Acharjee : Undergraduate Electrical Engineering | McGill University

Shovan was the photographer for the project. He took many pictures in different conditions, from all around McGill and Montreal for the purpose of testing the models. These pictures were shown in the presentation, and some are also included in this report.

Contents

1. Introduction

Autour is an eyes free mobile system that is designed to help visually impaired get a better understanding of their surroundings. It is a download able app, that uses the built in sensors of a phone, such as camera and GPS to reveal information and directions to the users. These details are relayed to the user through audio, in the similar way a sign or visual cues would to a sighted user. It also provides more in depth information about a particular location, on request.

The purpose of this project is to built an image captioning system for Autour. The user's camera will take pictures at a certain time interval. The image will be uploaded to cloud, where a computer will access the images, run it through a Deep Learning Network, and produce a sentence, describing the objects in the image. Any text present in the image will also be recognized. These descriptions will be returned back to the phone, where they will be relayed to the user in audio form. The diagram below illustrates the high level communication path.

Figure 1: System Communication Path

The importance of this project is self explanatory. As of August 2014, there are an estimated 285 million visually impaired people worldwide [3]. Of these, approximately 90% live in developing countries. Therefore it is safe to assume a majority of these people do not receive the adequate medical care or support to lead a normal life. The availability of a free system such as Autour, with image captioning capabilities, would drastically improve the quality of their lives. Outside Autour, image captioning has many more applications. From autonomous vehicles to robots, this technology can virtually be applied to any device that has a camera. Implementing such a system would be a giant step towards creating Artificial Intelligent beings.

2. Background

2.1 Image Classification and Detection

For humans, it is easy to look at an image, and recognize the different objects and actions taking place in the image. Not only can we recognize, but our brains can create complex sentences that can describe every single detail in an image giving a literal meaning to the quote "a picture tells a thousand words". Currently, there are two benchmarks in the subject of computer vision:

- The Pascal Visual Object Classes
- The ImageNet Large Scale Visual Recognition Challenge (ILSVCR)

These are conferences that have been running annually since 2005, with two specific goals:

1. To create a dataset of images and their corresponding descriptions
2. To provide an annual competition for anyone to test out new algorithms or models.

There are two challenges in particular:

1. Detection: where in the image is the object located
2. Classification: does the image contain a certain object

The two sections below compares both the conferences in terms of the challenges.

2.1.1 Detection

As mentioned above, the goal of this challenge is to recognize objects from a number of visual object classes in realistic scenes. The contestants are provided with a training set to train their networks on; a validation set to test the trained network, and a testing set, where the results will be used to judge the different algorithms. The categories were chosen to reflect different factors such as object scale, level of image clutterness, average number of object instance, and several others. The table below compares the two conferences [5]:

Table 2.1: Object Detection Conference Comparison

		Pascal VOV	ILSVRC
Number of object classes		20	200
Training	Num Images	5717	456567
	Num Objects	13609	478807
Validation	Num Images	5823	20121
	Num Objects	13841	55502
Testing	Num images	10991	40152

2.1.2 Classification

Image Classification was specifically for the ILSVCR conference only. The training data, a the subset of ImageNet dataset contains 1000 categories and 1.2 million images. The datset for validation and testing contain 150,000 images, collected from online search engines such as flickr. A random subset of 50,000 images with labels are released for validation, and the rest are to be used for testing the algorithms in the contest.

2.1.3 Results

There has been some great outcomes from these annual meetings, and each year results improve. In 2011, the best contestant of ILSVRC saw a classification error rate of 25%. in 2012, a deep CNN received an error rate of 16%. And in the next few years, error rate dropped to a few percent. Thus, a quantifiable amount of work has been done in the subject of Neural networks, and using them to detect, classify and describe images.

2.2 Optical Character Recognition (OCR)

Optical Character Recognition is conversion of images with typed, handwritten or printed text into machine-encoded text. For visually impaired users, it is important to know if there is any textual information present around them. The method below describes a simple OCR implementation.

The input for the model was single numbers. The numbers were both colored and grey scale. The first step of an OCR is to grey scale the image. This helps to find areas of contrasting colors and edges of characters from the backgrounds.A simple data set was used for the training. The size of the training images are 8 x 8 pixels. The colors varied from black and white to colorful samples. The figures below are sample inputs and the corresponding outputs. The number on the left are the input. The images in the middle are their black and white equivalent. And the charts on the right are the corresponding confidence scores for the guesses.

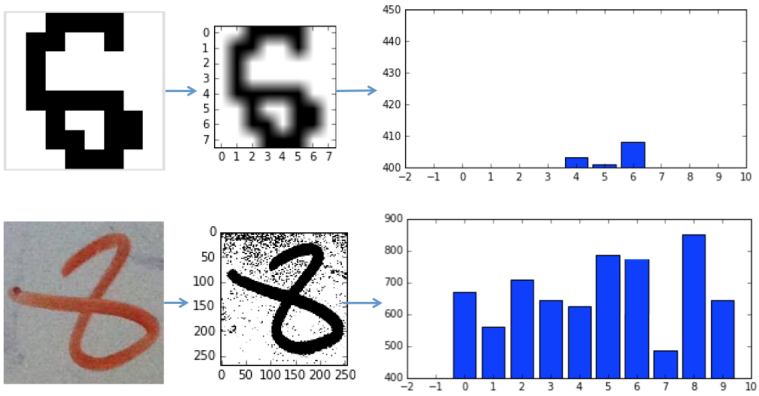


Figure 2: Input and output of a simple OCR model

3. Problem Representation

Images are made of two key ingredients:

- Objects
- Action

Objects are the different items present in an image. Actions connect the various objects in the image to one another. For example, the figure below shows a picture, the objects in the image (a dog) and the corresponding action (jumping). Feeding such an image through a caption generator should generate a description reading "A dog jumping".

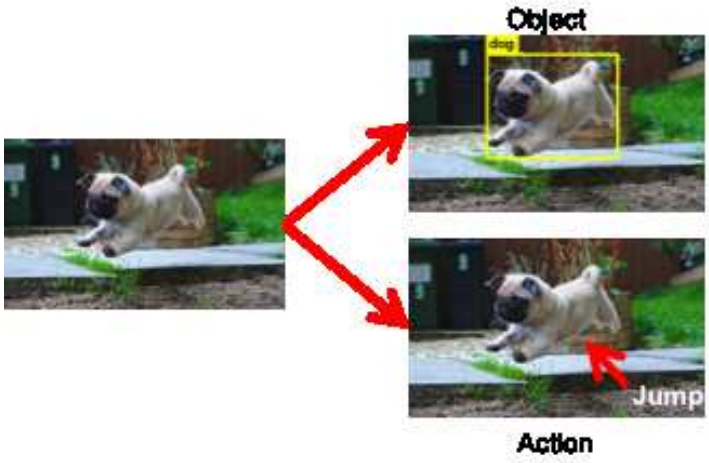


Figure 3: Objects (dog) and actions (jump) in an image

The task at hand is to create a model that can generate descriptions that not only captures these objects in an image, but also identify how the objects relate to each other, through attributes and activities.

A Deep Neural Network model will be created, and trained with images from data bases such as MSCOCO. Afterwards, images will be fed to the trained model, and the model will generate such description as the example above, and have a method of cross checking the captions to ensure accuracy.

For a visually impaired user, it is not enough to know what is around them, but also to know if they are any text that might be able to further identify their location and allow them to be more aware of their surroundings. As such, the model should be able to run Optical Character Recognition to identify any text that might be present. These labels could be street signs, building names, or even bus routes.

And finally, the user needs to be constantly aware of their surroundings. Thus the system is required to be time efficient, capturing images, analyzing them and relaying the information in audio format to the user at no slower than 10 second intervals.

4. Design and Implementation

4.1 High level overview

Summarizing from the problem representation section above, here are the system requirements and constraints for creating the automated Image Captioning model:

1. Generate a caption describing the image
2. Analyze and detect and text present in the image
3. Perform a cross check with a different neural network to validate the captions
4. Automate the system
5. Perform the analysis within a time constrain (within 10 seconds)

Based on the requirements above, a model with five components was created. These components are listed below, followed by a brief description of the operation:

1. OCR Detector
2. Caption Generator
3. Object Detection
4. Color Detection
5. Cross Checking

The OCR detector detects any text in the image. The caption generator is a deep neural network produces 3 captions (and their confidence scores) that best describes the image. The object detection uses a different deep neural network to identify and classify the objects present in the image. And the color detection lists the different colors found in the image. The cross checker uses the list of objects (and their synonyms) and colors to perform a linear search on the captions that were generated. It looks for the number of occurrences for these objects and the colors in each of the captions and uses the result and the caption confidence to output the final caption that best describes the image.

4.2 Model Implementation

The diagram below gives a visual description of the model containing each of the 5 components described above, the systematic path that connects each component, and an input and the its corresponding output.

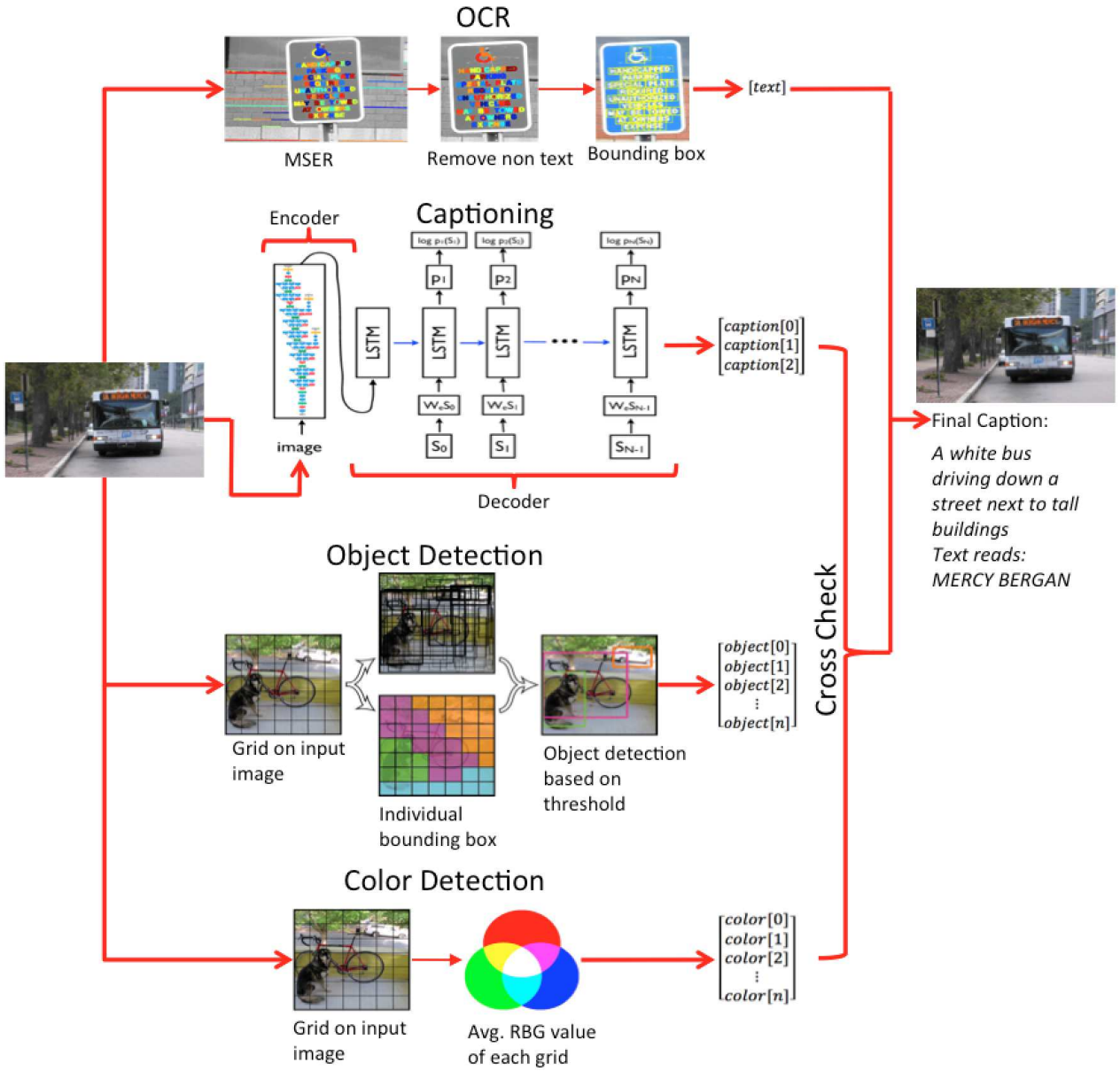


Figure 4: High level design implementation

The implementation of the individual components are as follows:

4.2.1 Optical Character Recognition

The importance of the text detection was outlined in the previous sections. Optical Character Recognition for this system was implemented using MATLAB. It has five key steps. [8] Before starting out, the input image is grey scaled in order to simplify the detections

The first step in the process is to identify the Text region using Maximally Stable External Regions. MSERs are connected areas characterized by almost uniform intensity, surrounded by contrasting background.They are used as a method for blob detection in images by finding correspondences between image elements from different viewpoints. The image below shows a grey scaled image with its MSER's highlighted.

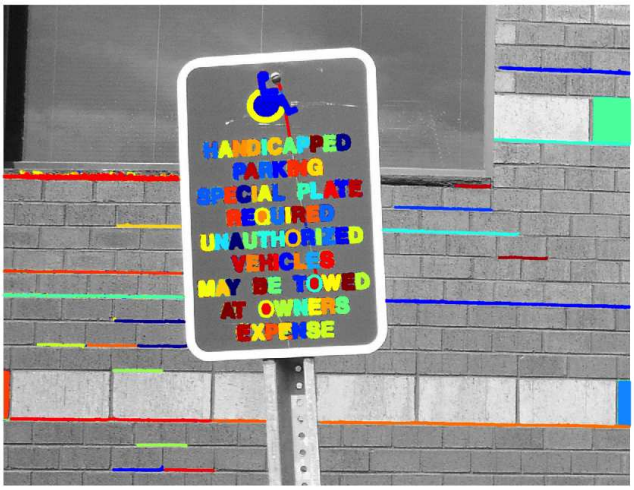


Figure 5: MSER's of a grey scaled image

The second step is to remove any non text regions. Observing the image above, we notice that the MSER algorithm detects most of the text in the image. But it also detects many other regions in the image that are not text (namely separation between bricks in the image). Thus the next step would be to remove these regions.

For the OCR model, this is done by examining the geometric properties of the regions, and comparing them to the geometric properties of text. These properties include Aspect Ratio, Eccentricity, Euler number, Extent, Solidity. Alternatively, a neural network can be trained to recognize these regions and exclude them. The image below shows the same example as the MSERs, but with the non test regions removed.

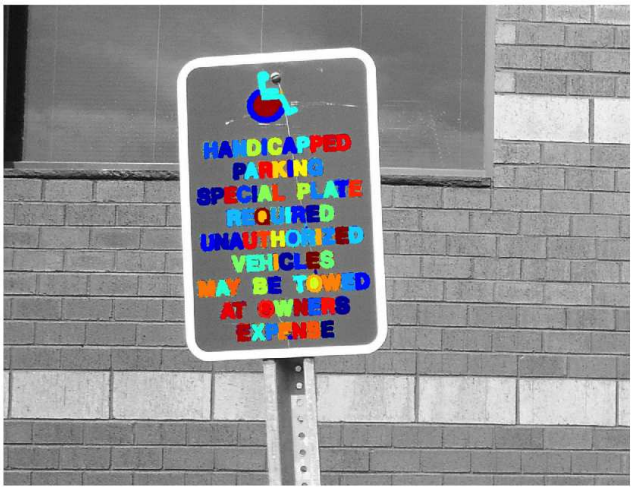


Figure 6: Image after the removal of non text areas

The third step for the program is to create bounding boxes around all the individual characters discovered. These bounding boxes identify the location as well as cover the entire character. At this point, it is possible to perform OCR on these bounding boxes, but the results would would be counter productive.

Indeed, it will be able to identify all the individual characters, but the chances of them being in an order that resembles the actual word are low.

For example, the image might contain the word 'hello'. Running OCR after all the individual characters were detected might return the characters in order, but chances are they will be returned in a random order, such as ['e','l','l','h','o']. Thus another step needs to be done before the image is ready to be analyzed. The figure below shows the image with all the individual bounding boxes.

Figure 7: Image with the individual bounding boxes

The fourth step is to combine all the overlapping bounding boxes. This step of the model solves the problem discussed in the previous step. This is accomplished by combining all the bounding boxes that overlap with one another to create a bigger bounding box. This allows the OCR to detect whole words or sentences rather than just the individual characters. The image below shows the output of combining all the overlapping bounding boxes.

Figure 8: Image with the overlapping bounding boxes combined

The fifth and final step is to perform OCR on the bounding boxes. After

detecting the text regions, the OCR function is used to recognize the text within each bounding box. Additionally, an error handler was included in the code, to handle specific errors caused by inputting images without text. In such cases, the model was hard coded to output 'No Text'.

4.2.2 Caption Generator

For the purposes of this paper, we propose a neural and probabilistic framework to generate captions for images. We decided to implement the model similar to the one proposed by Vinyals st al.[33]. It is a single joint model that takes an image I as input, and is trained to maximize the likelihood $p(S|I)$ of producing a target sequence of words S = $\{S_1,S_2,...\}$where each word S_t comes from a given dictionary, that describes the image adequately [33]. The neural network model is a two part model, known as encoder-decoder. Figure below shows a high level model architecture, with both the parts.

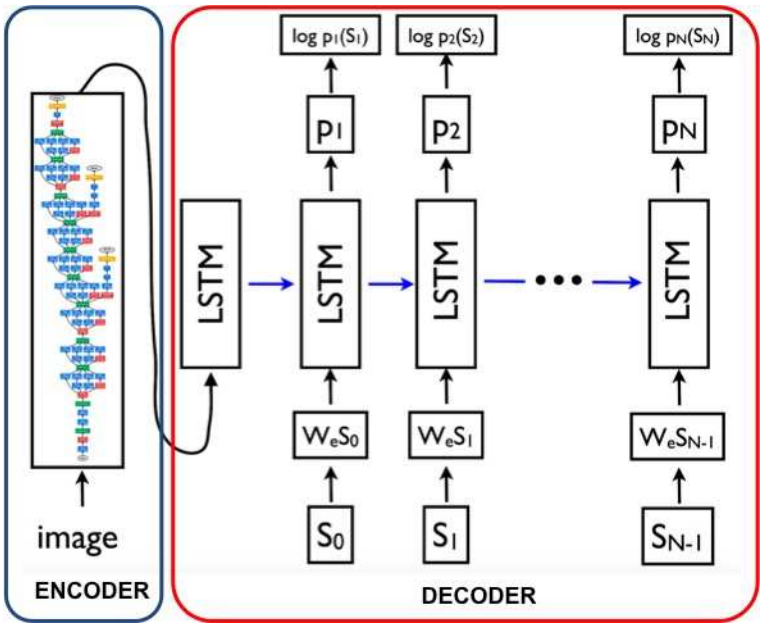


Figure 9: Caption Generator model

Encoder in the first part of the model. The input to the encoder is the image for which we want the caption. The encoder is a Inception V3 deep Convolutional Neural Network with 48 layers. Over the last few years it has been convincingly shown that CNNs can produce a rich representation of the input image by embedding it to a fixed-length vector, such that this representation can be used for a variety of vision tasks, mainly object detection and image classification. For this project, we use a CNN that is pre-trained on a data set to perform image classification, and using the last hidden layer as an input to the decoder stage of the model.

The decoder in the second stage of the model. The input comes from the previous layer in form of fixed length vectors that represent the image. The decoder is a LSTM (Long short-term memory) unit. This type of model is usually used for sequence modelling, such as speech recognition and language translation. The LSTM network is trained as a language model conditioned on the image en-

coding received from the encoder stage.

Words in the captions are represented with an embedding model. Each word in the vocabulary is associated with a fixed-length vector representation that is learned during training. In this diagram, $\{S_0, S_1, ..., S_{N-1}\}$ are the words of the caption and $\{W_e S_0, W_e S_1, ..., W_e S_{N-1}\}$ are their corresponding word embedding vectors [33]. The outputs $\{p_1, p_2, ..., p_N\}$ are probability distributions generated by the model for the next word in the sentence. The terms $\{logp_1(S_1), logp_2(S_2), ..., logp_N(S_N)\}$ are the log-likelihoods of the correct word at each step [33].

The model used for the project uses a beam search to generate the captions. The caption is generated in a word-by-word sequence, where at time step t, the caption generated by using the sentence already generated at time step t-1. The model produces three captions that best represents the image, along with confidence values for each of those captions.

4.2.3 Object Detection

"The contents of this section are heavily drawn from section 3.3 in report [10]"

Object Detection is the third component of the model. It is created using a CNN with 32 layers. It is trained on the COCO dataset using CUDA and darknet. The Figure below shows a high level process of how the detector works.

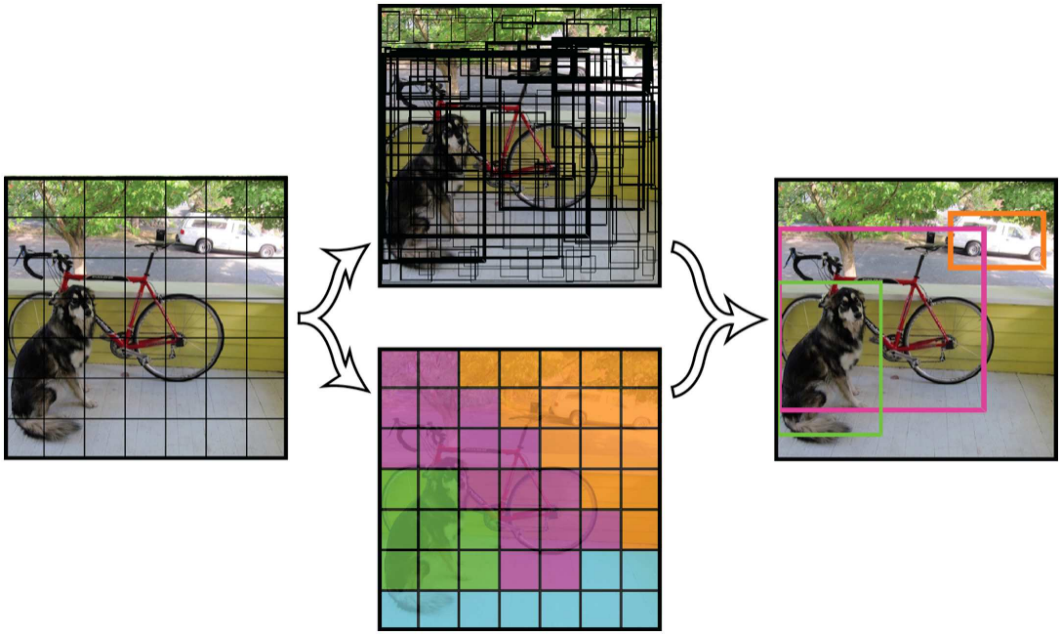


Figure 10: High level process flow of the object detection model

First a grid is overlaid on top of the input image, as shown in the left-colomost image in Figure 8. Each cell in the resulting image is responsible for predicting a few different things. Firstly, through edge detection and color detection, each cell is responsible for predicting bounding boxes around the object in its square. The cells also produce confidence values for the bounding boxes. This confidence score is the probability that bounding box contains the object. Grid cells with no objects produce low confidence scores, whereas grid cells with an object score high confidence values. When all the bounding boxes in the image

are displayed, a image similar to the one in the top picture of the middle column is produced. This is a map of all the objects present in the image, ranked by their confidence values. At this point the locations of the objects are known, but not their classification.

Next, each cell predicts some class probabilities. This produces a coarse segmentation map of the image, as shown by the image in the bottom of the middle column. Then the model performs an image classification and produces a conditional probability of an object existing in a certain segment. For example the segment on the top right, containing a car, the model is not saying with absolute certainty that there is a car, but rather if there is an object in that area of the image, then that object is most likely a car.

The model then takes these conditional probabilities and multiplies them with the confidence values generated earlier, which results in the generation of bounding boxes weighted by the actual probabilities for containing that object. This is shown in the figure below. A threshold is set, and all confidence values below the threshold are ignored, and the resulting image is the final image with the objects and their locations identified.

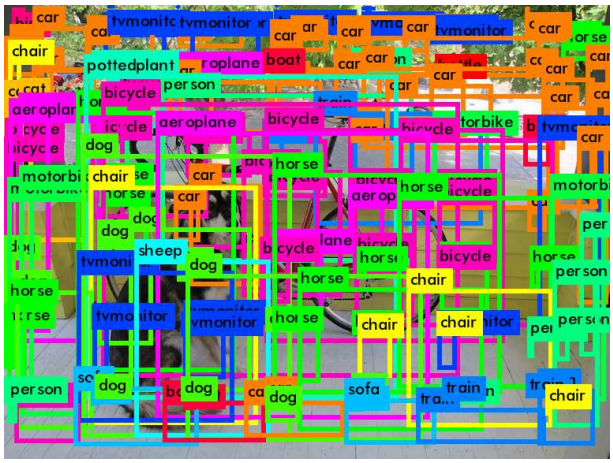


Figure 11: Image with all the ranked bounding boxes

4.2.4 Color Detection

Color detection is the fourth component of the model. It uses the RBG values from the image, and uses them to generate a list of colors present in the image. RBG stands for Reg, Blue, Green, and each of their indices vary from 0 to 255. Any combination of these three values produces an unique color, with white (presence of all colors) and black (absence of all colors) being at the extremes of the spectrum.

The Color Detector for the model is simple. The input image is broken up into small grids. The average RBG values are calculated for each grid.These values are then passed through a 3D range and placed in one of 11 common color groups. These color groups consist of the extremes, the primary and secondary colors as well as a few from the tertiary color group, as shown in the diagram below. The groups are: blue, red, green, pink, purple, black, white, grey, yellow, orange, brown. Finally, the model outputs a list of the individual colors found in

the image.

Figure 12: Color groups

4.2.5 Cross Check

Cross checking is the final component of the model. It takes the outputs from all the other components (except OCR) and outputs the best caption that describes the image. The steps performed to cross check the results are outlined below.

First, it takes the outputs from the Caption Generator and both the Object and Color Detectors. This includes:

- Three captions and their confidence scores
- A list of objects detected
- A list of colors detected

The second step is generation of synonyms from the results of object detection. This step is performed to create a more generalized list of objects. An example: the caption generator detects a woman in the image, which the object detector detects as a person. But since it does not have gender differentiation capabilities unlike the caption generator, the raw list of objects will produce 0 matches when cross checked. So therefore, the synonym generator is introduced. For person it generates a list which includes both man and woman. Thus, the updated list will now match the 'woman' in the caption.

The third step takes the list of objects (and the synonyms) and the list of colors detected, and performs a linear search through the three captions. It records the number of matches each captions produce.

The final step of the cross checker is to use both the confidence scores and the number of occurrences to calculate the best possible caption. This is done by simply multiplying the scores with their respective number of matches, and the caption with the highest pair is picked to represent the image.

5. Results and Tests

Testing of the model was done in two parts. First each of the four units were tested individually. Pictures in the test set were a mixture of images from around McGill and Montreal, as well as from online. Training dataset was Flickr8K, which included 10,000 images. The results and observations recorded from these tests are discussed below.

5.1 Optical Character Recognition

OCR was tested with various images of text under different constrains. These include distance to object, lighting, angle of text from camera and to test the error handling functionality, some images without text. The result of some of these images, as well as the image used to describe the model implementation are shown below. A few samples are shown below.

HANDICIXPPED
PARKING
SPECIAL PLATE
REQUIRED
UNAUTHORIZED
VEHICLES
MAY BE TOWED
AT OWNERS
EXPENSE

Figure 13a: OCR test: Input and Output

For the convenience of fellow guests,
please keep this area clean.

Par yard aux autres voyageurs,
veuillez conserver cet endroit propre.

Figure 13b: OCR test: Input and Output

MERCY BERGAN

Figure 13c: OCR test: Input and Output

NO TEXT

Figure 13d: OCR test: Input and Output

5.2 Caption Generator

The caption generator was tested simply by inputting a few images with a variety of items, and checking to assure that both objects and their actions were detected. The confidence scores were also noted. Below are a few test examples.

0) a group of people sitting in a room . (p=0.001519)
1) a group of people sitting around a table with laptops . (p=0.000280)
2) a group of people sitting around a table with a laptop . (p=0.000061)

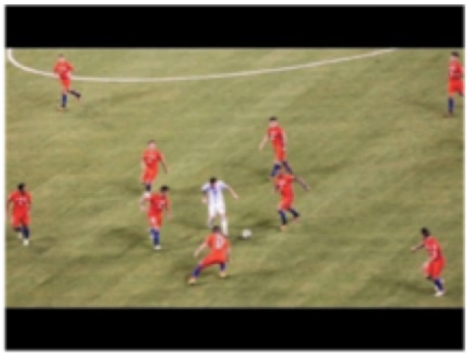
Figure 14a: Captioning Generator test: Input and Output

0) a city street with cars parked on the side of the road (p=0.000030)
1) a city street with cars parked on the side of the road . (p=0.000025)
2) a city street with cars parked on the side of it (p=0.000022)

Figure 14b: Captioning Generator test: Input and Output

0) a kid on a beach with a frisbee . (p=0.003592)
1) a man on a beach with a frisbee . (p=0.001153)
2) a person on a beach with a frisbee (p=0.000873)

Figure 14c: Captioning Generator test: Input and Output



0) a group of people on a field playing soccer . (p=0.003587)
1) a group of people playing a game of frisbee . (p=0.003430)
2) a group of people playing soccer on a field (p=0.001966)

Figure 14d: Captioning Generator test: Input and Output

5.3 Object Recognition

Object Detection model was tested by simply inputting the image, and observing how many objects were correctly identified. It has a threshold. Any objects with confidence scores below the threshold will be ignored. For the purposes of this project, the threshold was set to 0.1. Thus any objects with over 10% confidence score will detected. Below are a few examples, as well as the final output from figure 11.

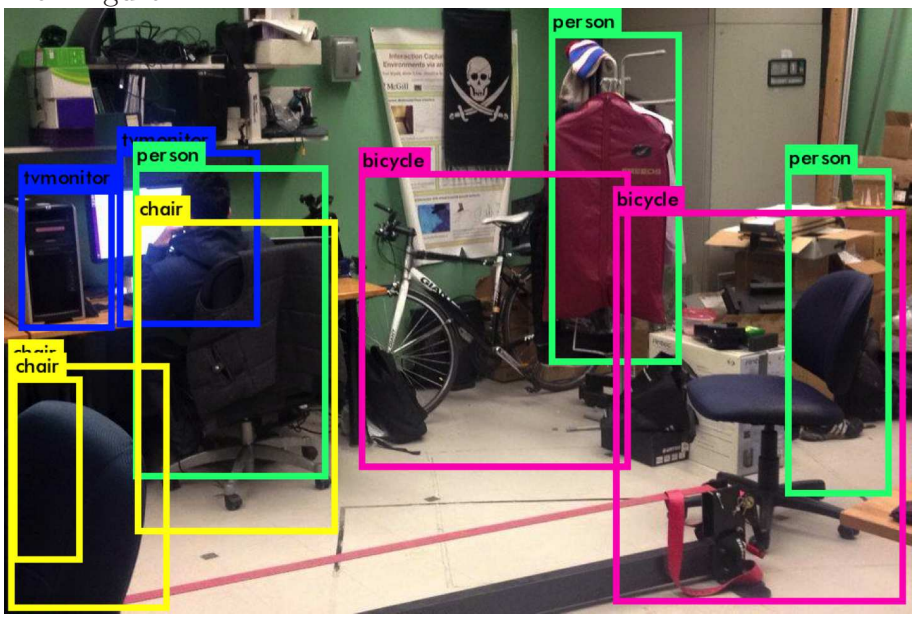


Figure 15a: Object detection result of Figure 11

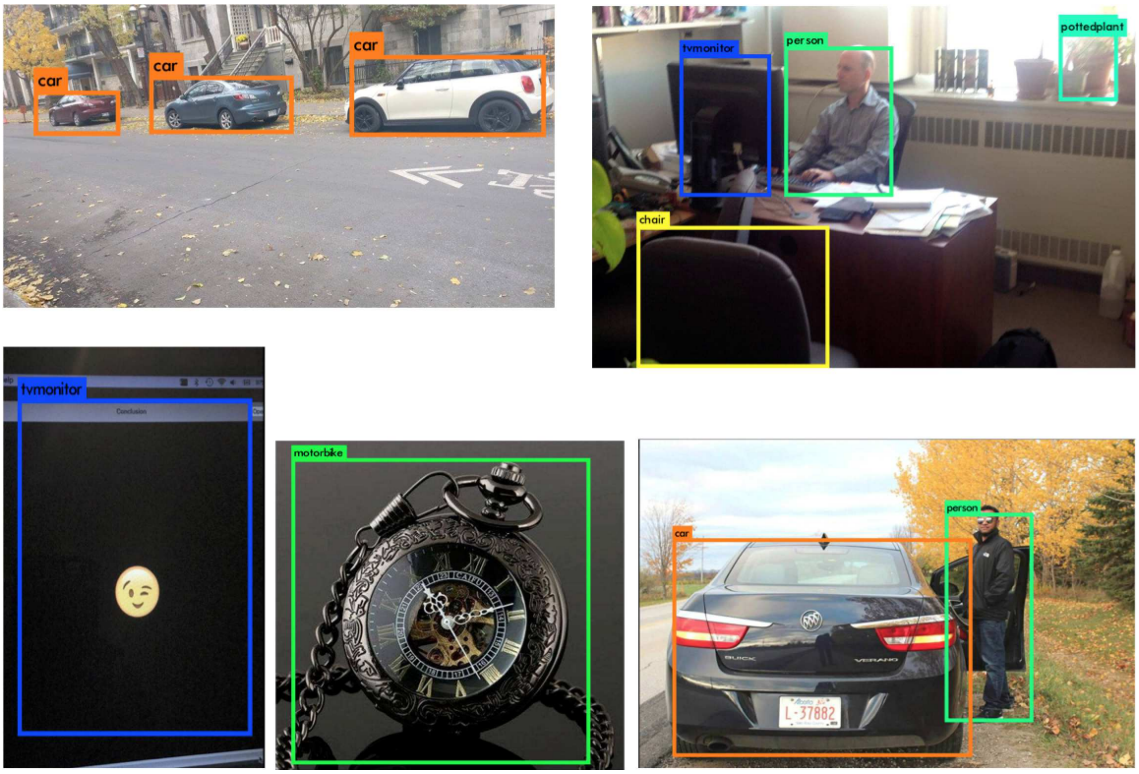


Figure 15b: Object detection result of sample images

5.4 Color Detection

as was the case for all the previous components, color detection was also done by inputting the image, and observing the results generated. To avoid duplicates in the results, the list was inserted in a set. Images with a variety of colors from the group of 11 were used for testing to ensure the accuracy of the results. Below are some inputs and outputs from the Color Detector.

Figure 16a: Color Detection test: Input and Output

Figure 16b: Color Detection test: Input and Output

Figure 16c: Color Detection test: Input and Output

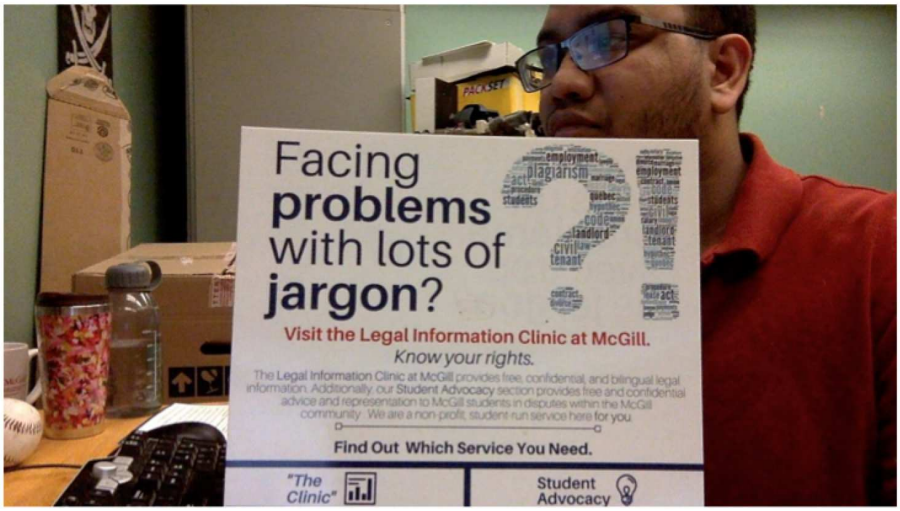
5.5 Integration Test

The Integration Test is the testing of all the individual components together to get the final output. The outputs from the latter three components were fed into the cross checker to get the final description. The test was performed on images with and without text, as well as a variety of objects and actions. A few examples are shown below.

Figure 17a: Integration test: Input and Output

A giraffe and zebra are in a field
Text reads: NO TEXT

Figure 17b: Integration test: Input and Output



A man holding a sign
Text reads: With lots of : jargon? Visit the Legal Information Clifiic at McGil Know your rights‘ Find Ou I Which Service You Need. Student < Advocacy

Figure 17c: Integration test: Input and Output

A street sing on a pole on a street

Text Reads: RUE JEANNE MANCE

Figure 17d: Integration test: Input and Output

6. Impact on Society and the Environment

This section analyses some of the costs, benefits, and potential risks in the implementation of this project as well as the societal impact that it imposes.

As mentioned earlier, there are over 285 million visually impaired individuals all over the world, out of which almost 40 million are completely blind. And a majority of this group are from low income families in developing countries, and thus have limited access to resources to aid in their impairment. These people also have to heavily rely on others on a daily basis. Thus providing them with an app such as Autour with its image captioning capabilities will not only improve their lifestyle, but also help them to achieve a higher level of independence.

6.1 Cost

6.1.1 User

Since this app is to be free of charge, the only cost for an user would be to purchase a phone with image taking capabilities, and to purchase a phone plan with data, for uploading the images for analysis.

6.1.2 Developer

On the developer end, the only expense was the purchase of a high end GPU for training the neural networks for the images. Most of the components used in the software model were open source, so no expense occurred in creating the program.

6.2 Safety and Risk

From the users point of view, the major concern should be to not rely solely on the app to tackle busy streets, as this might raise safety concern, for both the users and anyone around them.

Another safety concern is if the communication path is hacked, the users movements can be tracked at all times, since the phone is constantly taking images of the users surroundings. This issue will create a complete invasion of a users privacy, and take stalking to a different level.

However, the main safety concern is adversarial attacks. This happens when there is a noise introduced to the images before it is analyzed. This noise is virtually undetectable to the human eye, but when ran through the recognizer, the program outputs a wrong caption with high confidence. If the neural network is attacked, and this noise is introduced into the images, this would have a devastating effect on the users.

7. Conclusion

Throughout the course of this, we set out to create an image captioning system with text recognition capabilities. Several different papers were consulted. Many programs were implemented and tested on several images from online, as well as images from around Montreal and McGill.We have indeed created a system that is capable of describing images and text. Based on the results, we can see that the system does indeed do as required.

However, we haven't been able to fulfill all the goals set. The system does exactly as outlined, but there are still room for improvements. The time required per analysis is a little more than provided. Also the texts detected are not always accurate. The OCR works best with high quality images. Medium and low quality images require more detailed analysis, which consumes time. Further fine tuning is required to extract perfect results. Also, the code needs to be modified to allow the four components to run in parallel to generate the descriptions faster. And finally, the images and their descriptions should be stored and added to the training set, and the neural network programmed to train itself on the new images, for a set number of image caption pair stored.

Overall, this was an extremely interesting project to work on, and it has the potential to help millions of people around the world, and can help us take a giant leap towards Artifial Intelligence.

Bibliography

[1] Autour. (2016, July 29). Retrieved September 05, 2016, from http://autour.mcgill.ca/en/

[2] Blum, J. R., Greencorn, D. G., Cooperstock, J. R. (2012, December). Smartphone sensor reliability for augmented reality applications. In International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services (pp. 127-138). Springer Berlin Heidelberg.

[3] Visual impairment and blindness. (2014, August). Retrieved April 7, 2017, from http://www.who.int/mediacentre/factsheets/fs282/en/

[4] MathWorks. (n.d.). OCR. Retrieved February 5, 2017, from https://www.mathworks.com/help/vision/ref/ocr.html

[5] Large Scale Visual Recognition Challenge 2014 (ILSVRC2014). (2014). Retrieved April 1, 2017, from http://image-net.org/challenges/LSVRC/2014/indexintroduction

[6] Large Scale Visual Recognition Challenge 2015 (ILSVRC2015). (n.d.). Retrieved April 11, 2017, from http://image-net.org/challenges/LSVRC/2015/results

[7] Visual Object Classes Challenge 2012 (VOC2012). (n.d.). Retrieved April 1, 2017, from http://host.robots.ox.ac.uk/pascal/VOC/voc2012/

[8] Documentation. (n.d.). Retrieved February 2, 2017, from https://www.mathworks.com/help/vision/examples/recognize-text-using-optical-character-recognition-ocr.html

[9] Blum, J. R., El-Shimy, D., Bouchard, M., Cooperstock, J. R. Rendering the world to blind people via spatialized audio.

[10] Bashar S.(2016). Image Caption Generator. Honors Thesis 1, ECSE 498.

[11] Karpathy, A., Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3128-3137)

[12] Hendricks, L. A., Venugopalan, S., Rohrbach, M., Mooney, R., Saenko, K., Darrell, T. (2015). Deep compositional captioning: Describing novel object categories without paired training data. arXiv preprint arXiv:1511.05284.

[13] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2015). You only look once: Unified, real-time object detection. arXiv preprint arXiv:1506.02640.

[14] Chen, X., Fang, H., Lin, T. Y., Vedantam, R., Gupta, S., Dollár, P., Zitnick, C. L. (2015). Microsoft COCO captions: Data collection and evaluation server. arXiv preprint arXiv:1504.00325.

[15] Chen, X., Lawrence Zitnick, C. (2015). Mind's eye: A recurrent visual representation for image caption generation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2422-2431).

[16] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2625-2634).

[17] Karpathy, A., Joulin, A., Li, F. F. F. (2014). Deep fragment embeddings for bidirectional image sentence mapping. In Advances in neural information processing systems (pp. 1889-1897).

[18] Vinyals, O., Toshev, A., Bengio, S., Erhan, D. (2016). Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[19] Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A., Murphy, K. (2015). Generation and comprehension of unambiguous object descriptions. arXiv preprint arXiv:1511.02283.

[20] Andreas, J., Rohrbach, M., Darrell, T., Klein, D. (2016). Neural module networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 39-48).

[21] Reed, S., Akata, Z., Schiele, B., Lee, H. (2016). Learning Deep Representations of Fine-Grained Visual Descriptions. arXiv preprint arXiv:1605.05395.

[22] Akata, Z., Malinowski, M., Fritz, M., Schiele, B. (2016). Multi-Cue Zero-Shot Learning with Strong Supervision. arXiv preprint arXiv:1603.08754.

[23] Zitnick, C. L., Dollár, P. (2014, September). Edge boxes: Locating object proposals from edges. In European Conference on Computer Vision (pp. 391-405). Springer International Publishing.

[24] Blaschko, M. B., Lampert, C. H. (2008, October). Learning to localize objects with structured output regression. In European conference on computer vision (pp. 2-15). Springer Berlin Heidelberg.

[25] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., Zisserman, A. (2010). The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2), 303-338.

[26] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Berg, A. C. (2015). Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 115(3), 211-252.

[27] Image Recognition. (n.d.). Retrieved October 05, 2016, from https://www.tensorflow.org/versions/r0.12/tutorials/image_recognition/index.html

[28] Guerzhoy, M. (n.d.). CSC321: Introduction to Machine Learning and Neural Networks (Winter 2016). Retrieved September 15, 2016, from http://www.cs.toronto.edu/ guerzhoy/321/

[29] Nielsen, M. (2016, January). Neural networks and deep learning. Retrieved September 15, 2016, from http://neuralnetworksanddeeplearning.com/

[30] Britz, D. (2015, October 08). Recurrent Neural Networks Tutorial, Part 3 – Backpropagation Through Time and Vanishing Gradients. Retrieved September 15, 2016, from http://www.wildml.com/2015/10/recurrent-neural-networks-tutorial-part-3-backpropagation-through-time-and-vanishing-gradients/

[31] O. (2016, May 14). Retrieved September 15, 2016, from https://www.youtube.com/watch?v=AQirPKrAyDg

[32] C., Olah. (2015, August 27). Understanding LSTM Networks. Retrieved September 15, 2016, from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[33] Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

[34] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on (pp. 248-255). IEEE.

[35] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Baby talk: Understanding and generating simple image descriptions. In CVPR, 2011.

[36] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014

[37] I. J. Goodfellow, J. Shlens, and C. Szegedy.Explaining and Harnessing Adversarial Exam-ples.ArXiv e-prints, December 2014.

[38] Abigail Graese, Andras Rozsa, and Terrance E.Boult.Assessing threat of adversarial ex-amples on deep neural networks.CoRR,abs/1610.04256, 2016.

[39] Sandy H. Huang, Nicolas Papernot, Ian J.Goodfellow, Yan Duan, and Pieter Abbeel.Adversarial attacks on neural network policies.CoRR, abs/1702.02284, 2017

[40] Oriol Vinyals, Alexander Toshev, SamyBengio, and Dumitru Erhan.Show andtell:Lessonslearnedfromthe2015MSCOCO imagecaptioningchallenge.CoRR, abs/1609.06647, 2016.

Appendix

7.1 List of Abbreviations

DLN Deep Learning Network
NN Neural Network
CNN Convolutional Neural Network
RNN Recurrent Neural Network
BRNN Bi-Directional Recurrent Neural Network
mRNN Multimodal Recurrent Neural Networks
COCO Common Objects in Context
MSCOCO Microsoft Common Objects in Context
RGB Red Green and Blue
OCR Optical Character Recognition
sentids Sentence ID's
CIM Center for Intelligence Machines
VOC Visual Object Classes
ILSVRC ImageNet Large Scale Visual Recognition Competition
GPS Global Positioning System