Interim Report

Individual Undergraduate Project 2017/2018

- **Student:** Ovidiu-Horatiu Ilie, 3rd year student, Dept. of Computer Science, UCL

- **Project title:** Image to Speech – An Everyday Assistant for Blind People

- **Current project title:** Image to Speech – Building A Simple Assistant for Blind People

- **Supervisor:** Niloy J. Mitra, Professor of Geometry Processing, Dept. of Computer Science, UCL

- **Progress made to date:**
  During the first part of the project, a lot of progress has been made both within researching suitable methods that might be used and within designing and developing suitable solutions towards achieving the outcome.

  As a reminder, my project is supposed to deliver a way to translate smartphone captured images (and possibly videos) into accurate audio descriptions. Everything should be integrated into an easy-to-use phone application that can help blind people (and not only) to get correct descriptions of their surroundings, making a lot of their tasks easier.

  Now, probably the most important part of my project is to come up with a Machine Learning solution that can learn and deduce descriptions from images and translate them into words. Nowadays, there exist a lot of APIs implementing such features (e.g. Microsoft Computer Vision API or IBM Watson). However, me and my supervisor established that it would be better for me to create my own implementation, as it would show my understanding about the problem and it will allow me to evaluate how the model I created is behaving.

  So, in the first instance, I spent a lot of time learning more about Machine Learning and Convolutional Neural Networks, how they work and how I might use them. By meeting with my supervisor weekly, I refined my knowledge about the subject so that I can start working on the model itself. Moving on, I started reviewing papers related to my project, from classifying objects and semantic segmentation, to describing images in one go by connecting both a CNN and a RNN into one solution. These certainly helped me to understand what others have done before and how I should proceed next.

  Now, before actually implementing my ML solution, I had to actually find good datasets for training. After inspecting some possibilities, I decided to use NYU-v2 indoor dataset as my starting point for training – later, I can replace this if needed. Then, I moved into actually building the model. For this, I have used the TensorFlow library to build models and fit them to the data. Right now, my model is capable of identifying objects within images with good confidence and it shows the confidence value. Since the dataset is quite big, I am currently training my algorithm on a GPU to speed up the process. Due to several issues with my GPU, I did not manage to implement a good working version of the semantic segmentation which is needed.

Regarding the Android application I need to develop, I managed to create a basic UI for it and to create a photo capture system within it. At the moment, the app captures a frame and sends it to a remote server from where the predicted information is given and displayed as text. I chose to use a server for giving predictions, as my final model will be quite computationally expensive, and it won't be a good choice to let it run on a phone. Therefore, I have created a cloud VM where my model is automatically deployed using Git and I opened an endpoint where the model can be accessed for predictions.

- **Remaining work to be done:**
  At the moment, my model is not doing everything it is supposed to do within the final version of the app (see above). Therefore, until the end of the project, I need to make it work well so that it can give me detailed text descriptions from the server back to the Android application. At the moment, I am planning to finish my CNN model for segmentation and to combine it with a RNN for getting human-like descriptions of the data.

  Moving on, I will need to create and describe evaluation strategies of the model, so that I know how well different versions of my model compare to already existing methods. I think it would be helpful to have the evaluation scripts run automatically whenever I deploy a new model to the server, so that I know exactly the evaluation metrics straight away. Automated tests for both the server code and the Android app will be needed in the form of unit tests. Even if the final codebase will not be big, it is good to have them as a caution mechanism within the project. For security improvement, I am considering imposing the usage of a key when accessing the server (at the moment, anyone can make use of it by sending a request, which is not a good practice for the future).

  Regarding the mobile application, I need to improve the UI, to make it intuitive and as simple as possible. Once the model is completely done and I receive good text descriptions from the model on the server, a simple text-to-speech API will be used within the app in order to *pronounce* the text for the user. Moreover, I will try to make good predictions from video too, provided that this won't take too long to implement.

  If time allows, a cool feature would be to use speech recognition APIs in order to allow the user to control the app by voice. I think that automating simple tasks by voice (such as taking another picture or repeating the last description) will be a huge improvement, especially since the app is mainly focusing on helping blind people.

Finally, I certainly need to write up my report and all the documentation needed for my submission. Writing a specification document for my work is required for further use and further extensions of the application. If time allows, I am considering implementing my server model as a REST API, where people can request development keys and call the server when needed. If everything goes well, it might be important to make my service publicly available.