Image Recognition Challenge for Rooms (from Microsoft)
— Report One —

Tomasz Bartkowiak, Suampa Ketpreechasawat, Nattapat Chaimanowong,
Danlin Peng, Lin Li, Yini Fang
{tb2816, sk3217, nc17, dp2315, ll4117, yf3016}@doc.ic.ac.uk

Supervisor: Dr. Anandha Gopalan
Course: CO530, Imperial College London

27th January, 2018

1 Introduction

Visual information such as images play an important role in how people make decision on accommodation. A platform which provides accommodation should ideally organise data and images in such a way that it is simple and intuitive for the user to browse through. Each user will likely have a different set of criteria for choosing an accommodation. For example, a student might want to prioritise his accommodation choice based on the size of the available workspace. In this case the user should be able to configure the platform to prioritise or show only the images of the room of interest for each accommodation. With millions of images available, tagging and organising them manually is practically impossible. However, based on the recent advancement in Convolution Neural Networks (CNN)(1–6), the task of classifying images can now be carried out by computer software to a high degree of accuracy. The goal of the project is to create and deploy a proof of concept for the usage of CNN and Microsoft Cognitive Toolkit (CNTK) to organise images on an accommodation listing platform (web-based and/or mobile app). The approach of this project is to utilise Faster Region-based Convolutional Neural Network (Faster R-CNN)(6) method to determine which objects are presented in a given image and then, using this information, deduce the type of room depicted by the image. The project also aims to provide feedback to Microsoft on the usage of CNTK and, if possible, contributes to the open source community.

2 Requirements

2.1 Stakeholders

The stakeholders in the project are list below:

1. The user, which can be anyone who is looking for an accommodation. The project aim to make it easier for users to select accommodation by organising images to match each users need.
2. Microsoft. The project aim to provide Microsoft with the feedback on the usage of CNTK in implementing CNN. This will allow Microsoft to improve CNTK even further.
3. The open source community. From the experience and insights gained during the project, contribution can be made to the open source community, especially relating to the topic of CNN and CNTK.

2.2 Minimum Requirements

The minimum specification for the project is the deployment of an accommodation listing platform which utilises CNN to organise images based on users preference. For the minimum viable

product (MVP), the platform will be deployed as a web application. In the MVP, the platform will allow the user to search and filter accommodations based on common metric such as location and price range. The platform will organise the search results to show only images of the specified room for each accommodation (if the user choose to do so). The MVP will also allow accommodation providers to post details and images for new accommodations to the platform. These images will be automatically tagged and organised using the trained CNN room classification model. To support the web application, a RESTful API containing the trained CNN model and a database for accommodations must be deployed. Additional specification from Microsoft is for the model to be implemented and trained using CNTK and for the RESTful API to be deployed as an Azure Web App.

The requirements from the minimum specification of the project are summarised below in order of priority (highest to lowest). The requirements are also categorised according to 4 categories : Essential/Non-Essential, and Functional/Non-Functional

1. Implementation and deployment of a web application that supports all the functionality in the specification. (Essential, Functional)
2. Deployment of RESTful API containing the CNN room classification model and database as an Azure Web App (Essential, Functional)
3. Creation and training of model in CNTK. Gathering any data needed to train the model is a part of this requirement. (Essential, Non-Functional)
4. Model should performed the specified classification task up to reasonable accuracy. (Essential, Functional)

2.3 Advanced Requirements

The possible extension that could be added to the project once the minimum requirements are completed are considered in this section. These extensions are listed below in order of priority (highest to lowest) and categorised using the same set of category as the minimum requirements.

1. Improve model accuracy by utilising more advanced CNN architecture such as ResNet(7). (Non-Essential, Non-Functional)
2. Additional features. One possibility could be allowing the user to sign in and save accommodations into their shortlist. Another possibility is to suggest similar accommodation based on other accommodations that the user has shortlisted. (Non-Essential, Functional)
3. Additional deployment platform. This could be in iOS or Android and should improve the user experience for the platform on mobile devices. (Non-Essential, Non-Functional)
4. Improve model evaluation time by using alternative method of object detection. Possible approach to consider is the Single Shot MultiBox Detector (SSD)(8). (Non-Essential, Non-Functional)
5. Comparison of accuracy between different models. This could involve comparing the accuracy against models which go from input image to category directly using a CNN without any separate identification of objects. (Non-Essential, Non-Functional)

3 Feasibility and risk

Every project has risk associated with it, mainly due to the time- and human capital- constraints. It is therefore of huge importance to qualify, and quantify (in terms of risk value) every projects task and requirement and estimate all needed resources to avoid delays, providing unfinished/faulty product/service or not providing it at all. The risk shall be categorised according to two categories: Harmfulness (Low, Medium, High) if the requirement is not met in due time, and feasibility (Low, Medium, High) - stating how probable it is to achieve satisfactory results.

3.1 Minimum Requirements

1. Implementation and deployment of a web application that supports all the functionality in the specification.

   (a) Risk associated: Web interface is not created, does not allow for sufficient user interaction. or does not meet all requirements as mentioned in the specification.
   Harmfulness: High

   (b) Feasibility: Medium, due to the abundance of sources concerning Web application development, creating a simple interface is feasible within 1-2 weeks. Providing sufficient user experience is however a harder task that requires more time and people engaged in the development.

2. Deployment of RESTful API containing the CNN room classification model and database as an Azure Web App

   (a) Risk associated: API is not created or does not work properly for the external user.
   Harmfulness: High

   (b) Feasibility: High, the basic skeleton of the API has been already provided and tested on our separate VM for Web application.

3. Creation and training of model in CNTK. Gathering any data needed to train the model is a part of this requirement.

   (a) Risk associated: The model cannot be created (software incompatibility or hardware restraints) or there is a lack of data.
   Harmfulness: High

   (b) Feasibility: High, the basic model has been already created by the Microsoft team and tested by us on Azure Virtual machine. Transfer learning will be used to take advantage of the data that has already been analyzed and labelled.

4. Model should perform the specified classification task up to reasonable accuracy.

   (a) Risk associated: The model does not classify images or does it with low accuracy.
   Harmfulness: High

   (b) Feasibility: High, although the basic model is not accurate enough, it has been tested with success and the team is working on developing it with provided tools.

3.2 Advanced requirements

All advanced requirements have a default harmfulness of Low

1. Improve model accuracy by utilising more advanced CNN architecture such as ResNet.

   (a) Risk associated: The model is not improved.

   (b) Feasibility: Medium. Depending on how the project progresses there might be time for architecture improvement.

2. Additional features. One possibility could be allowing the user to sign in and save accommodations into their shortlist. Another possibility is to suggest similar accommodation based on other accommodations that the user has shortlisted.

   (a) Risk associated: The service provides just baseline functionality without any features.

   (b) Feasibility: High. Implementation of those features is feasible given current time constraints.

3. Additional deployment platform. This could be in iOS or Android and should improve the user experience for the platform on mobile devices.
   (a) Risk associated: Mobile application is not deployed.
   (b) Feasibility: High, a simplified version of iOS app has already been created and allows the client to make a photo of a room, giving him some (although not really accurate) results concerning objects recognition using pretrained network.
4. Improve model evaluation time by using alternative method of object detection. Possible approach to consider is the Single Shot MultiBox Detector (SSD).
   (a) Risk associated: The model is not improved.
   (b) Feasibility: Low, Further model improvements require more time to give satisfactory results.
5. Comparison of accuracy between different models. This could involve comparing the accuracy against models which go from input image to category directly using a CNN without any separate identification of objects.
   (a) Risk associated: No other model has been improved and therefore model comparison is impossible.
   (b) Feasibility: Medium, Even with some slight different approaches/models tested, comparison in terms of accuracy can be made.

4 Technical Requirements

4.1 Development Environment

During the development stage,the model used for recognition will be built and trained on the Azure DSVM (Data Science Virtual Machine)[1] with a operating system of Linux Ubuntu 16.04 LTS. After finishing training, i.e. production stage, a web application will be deployed on the Azure Web Apps[2] to provide room recognition services through specifically designed RESTful API with a public website also hosted on the same server as a basic entrance to the aforementioned services.

4.2 Programming languages

Due to the requirement from baseline project pre-supplied by Microsoft, both of the Faster-RCNN model and web application will be developed using Python, specifically version 3.5. Besides, a group of front-end techniques, JavaScript, HTML and CSS, will be integrated to develop the final public product, i.e. currently the website.

4.3 Dependencies

4.3.1 Baseline Projects

The works of two parts, deep learning and web application, mentioned before will be developed respectively based on the baseline projects(9)(10) provided by the Microsoft, although the team is also considering to work from scratch after obtaining enough inspiration and experience from baseline projects.

4.3.2 Transferred Model

Due to the consideration of efficiency, the amount of parameters need to be trained is around 50 million, some filters and weights will be, using technique of Transfer Learning, extracted from existing pretrained model AlexNet(1).

1. Data Science Virtual Machine for Linux (Ubuntu): https://azuremarketplace.microsoft.com/en-us/marketplace/apps/microsoft-ads.linux-data-science-vm-ubuntu
2. Azure Web Apps: https://azure.microsoft.com/en-gb/services/app-service/web/

4.3.3 External Frameworks/Libraries

Basically, the deep learning model, i.e. Faster-RCNN, will be achieved using recognitive tool kit of Microsoft named CNTK[3] 2.1 as required. Besides, the Anaconda[4] will be taken as a base due to its rich resources of data science packages, specifically NumPy, Pandas and Matplotlib etc., and powerful feature of virtual environment management. Apart from that, the part of web application and website will depend on the Flask module of Python and React library of JavaScript respectively.

4.4 Dataset

Currently, the only supplied data is from Hotailors[5] with a total number of 113 images for only two types of room, bedroom and bathroom, which obviously leaves a large space to improve the accuracy. During the next stage, some new available datasets, like ImageNet[6], will be explored.

4.5 Version Control System

Git is selected to help us manage the whole development, like the merging of different programmers work and the management of different version of code. Another consideration for using Git is the convenience of Gitlab, thanks to the support of DoC, to keep our project private without any extra cost before completion.

5 Development Strategy

5.1 Development Strategy Chosen

In regard to the work efficiency and time constraint of the project, the team members are allocated into 2 sub-teams - Deep Learning Team and Web Development Team, working in parallel.

1. Deep Learning Team focuses on extension of baseline DNN model with the aims to improve its robustness and accuracy.
2. Web Developement Team
   (a) Front-End Development involves graphical user interface design for web service.
   (b) Back-End Development emphasizes on the data import, databases and deployment of the DNN model on web service.

After the primary goals are satisfied, the rotation of team members will be considered as an alternative of code review and feedback as well as a way to encourage each individual to expand their expertises in different areas.

In order to systematically track work progress and convey sufficient communication between team members, Scrum is implemented as a practice for agile development. Sprint planning meeting is scheduled biweekly for preparing the lists of tasks on upcoming sprint. Daily scrum meeting will be conducted for individuals to briefly update their progress within their sub-team. At the same time, the team plans official meeting twice a week, aiming for all members to report any difficulties encountered and task achievement in respect to product backlogs. Plus, this session will provide the opportunity for brainstorming in which team members could have open discussion on each particular issue, and the team leader will be responsible for capturing and integrating those ideas from team members.

3. Microsoft Cognitive Toolkit(CNTK): https://www.microsoft.com/en-us/cognitive-toolkit/
4. Anaconda: https://www.anaconda.com/what-is-anaconda/
5. Hotailors: https://hotailors.com/
6. Imagenet: http://www.image-net.org

5.2 Development Plan

The figures below show a Gantt chart of this project, which clearly indicates all the tasks required with begin weeks and end weeks (the first week starts at 1st January of 2018). The colours of bars in the right-hand side indicate whether the tasks have been done already. The gantt chart will be changed frequently during this project according to the completion percentages of tasks. This gantt chart below show the current completion percentages of tasks at the beginning of week 5.

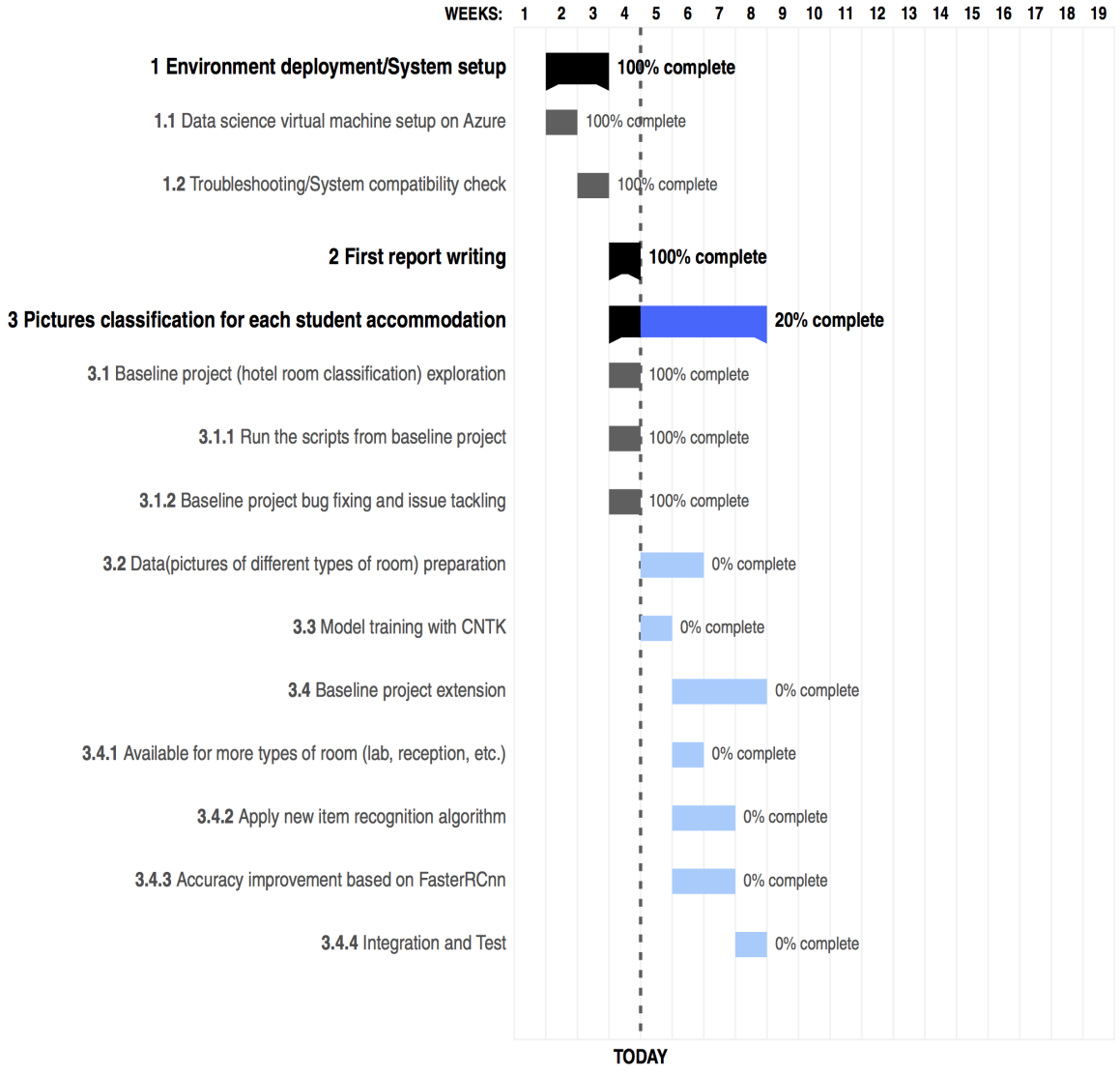

Figure 1: Gantt Chart

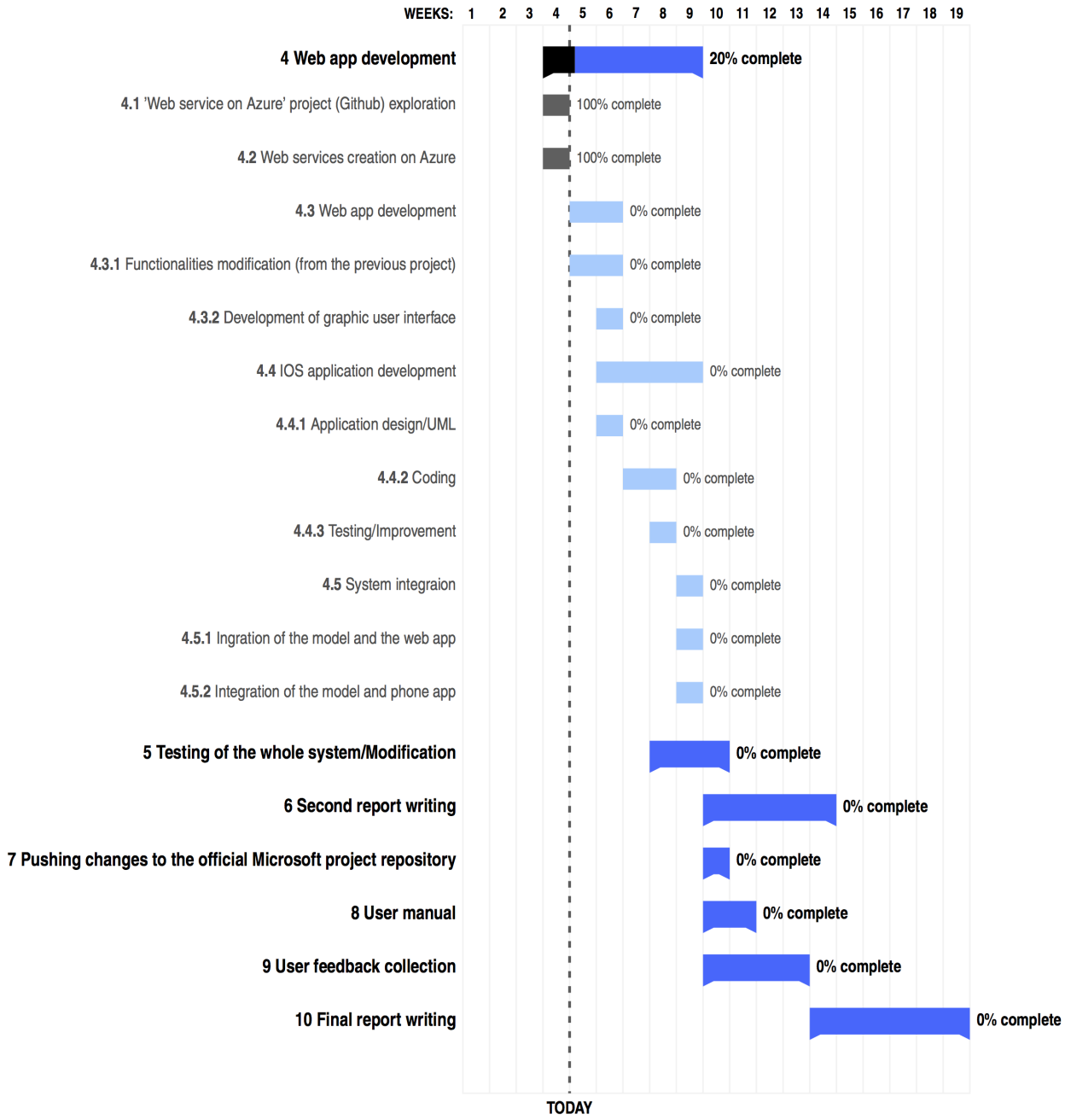

Figure 2: Gantt Chart(cont.)

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. *ImageNet Classification with Deep Convolutional Neural Networks*. Curran Associates, Inc., 2012.

[2] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014.

[3] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. *Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation*. IEEE Computer Society, Washington, DC, USA, 2014.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition*. Springer International Publishing, Cham, 2014.

[5] Ross Girshick. Fast r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[6] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[8] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016.

[9] Karol Zak. Cntk-hotel-pictures-classificator. `https://github.com/karolzak/CNTK-Hotel-pictures-classificator`, 2017.

[10] Karol Zak. Cntk-python-web-service-on-azure. `https://github.com/karolzak/CNTK-Python-Web-Service-on-Azure`, 2017.