Bilkent University

Department of Computer Engineering

Senior Design Project

Prelude

High-Level Design Report

Group Members: Samet Özcan, Osman Orhan Uysal, Osman Burak İntişah, Mehmet Alper Karadağ, Ziya Erkoç

Supervisor: Uğur Güdükbay

Innovation Expert: Veysi İşler

Jury Members: Hamdi Dibeklioğlu, Shervin Rahimzadeh Arashloo

Analysis Report

May 22, 2020

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/1.

1. Introduction

1.1 Purpose of the system

In the textile industry, the clothes produced in some factories are manually examined in a cloth inspection machine to detect anomalies. When an anomaly occurs, the personnel operating that machine is responsible for either fixing the error right away or marking the improper section so that it can be replaced later. Yet, since people need to scan the cloth flowing continuously, they might miss some anomalies which leads to faulty cloths due to fatigue and the limitations of human nature, therefore, decrease in the companies' product prices and loss in their reputation [1, 2, 3].

1.2 Design goals

1.2.1 Usability

Our software should be easily used by textile experts since they race against time. So, they should be able to select defects quickly to not lose much time.

1.2.2 Performance

Our software should be able to detect defects on clothes fast since the cloth inspection machine runs quite fast. That is, our software should both incorporate high frequency cameras and find defects fast. If either one of them is slow, then some part of the clothes may not be captured or the defected part passes and textile expert would need to rollback the cloth inspection to locate the defect and fix it. Hence, we should find a fast neural network and high frequency camera.

1.2.3 Reliability

Our software should not frequently mislead the personnel by generating false positive or false negative predictions. That is, it should not consider perfect cloth as faulty or vice versa.

1.2.4 Security

Security of the image and label data is important in our project due to the policy of the company. Especially, our data collection software should not connect to the internet. Data will be temporarily moved to cloud server for training but this should be done in compliance with the firewall policy of the company.

1.3 Definitions, acronyms, and abbreviations

RCNN: Regional Convolutional Neural Network. It is a deep learning algorithm used for object detection.

1.4 Overview

Prelude is a cloth defect detection software that is aimed to be compatible with any cloth inspection machine. It also aims to detect defects rapidly using the fastest object detection algorithm to keep up with the race of every cloth inspection machine. Prelude is a complex software and consists of three modules. First module is to be used by textile experts who will label defects in the fabric images. We are going to develop a very simple user interface for them so that they can label quickly. Second module is the training module. In this module, we will train our deep learning model using the labeled images generated by the previous module. Third module is prediction module. This module concerns both textile experts (or the personnel operating the cloth inspection machine) and the managers of the company. Our deep network will be predicting the images coming from the camera feed using the weights acquired from trained model. The module will halt the inspection machine when it predicts a defect. At this time, expert may fix the defect or correct the model if the prediction is incorrect. In this module, managers can also generate detailed reports containing critical information such as which defects occur at which fabrics frequently. We also want to incorporate first module into third module because we want textile experts to label some defects if our model missed it. In other words, we want the textile experts to fix false negatives.

2. Current software architecture (if any)

Currently some products are developed to detect defects in clothes but we could not learn their underlying software architecture since they are not open source.

3. Proposed software architecture

3.1 Overview

We propose a whole system from data collection stage to detection stage. Our data collection provides an effective way for workers to label defects in a multiclassed manner to be used in training, validation and test phases later. Our proposed model is a deep learning model called Faster RCNN. Specifically, it is an improvement that is built upon RCNN which makes its predictions around 200 milliseconds [4]. Therefore, we find it suitable for our problem. Lastly, we also propose a comprehensive user interface where users can view statistical data about the system alongside the data collection tool

3.2 Subsystem decomposition

Our high level design comprises of 4 layers. The user interface layer is the top layer where the user interacts with our software. This layer have access to all other layers. Report layer encapsulates functionality related to detailed reports generated for the managers of the textile company to use. Machine Learning layer contains all neural network related logic and the dataset. Finally, the data layer contains data type classes that are used by all other layers.

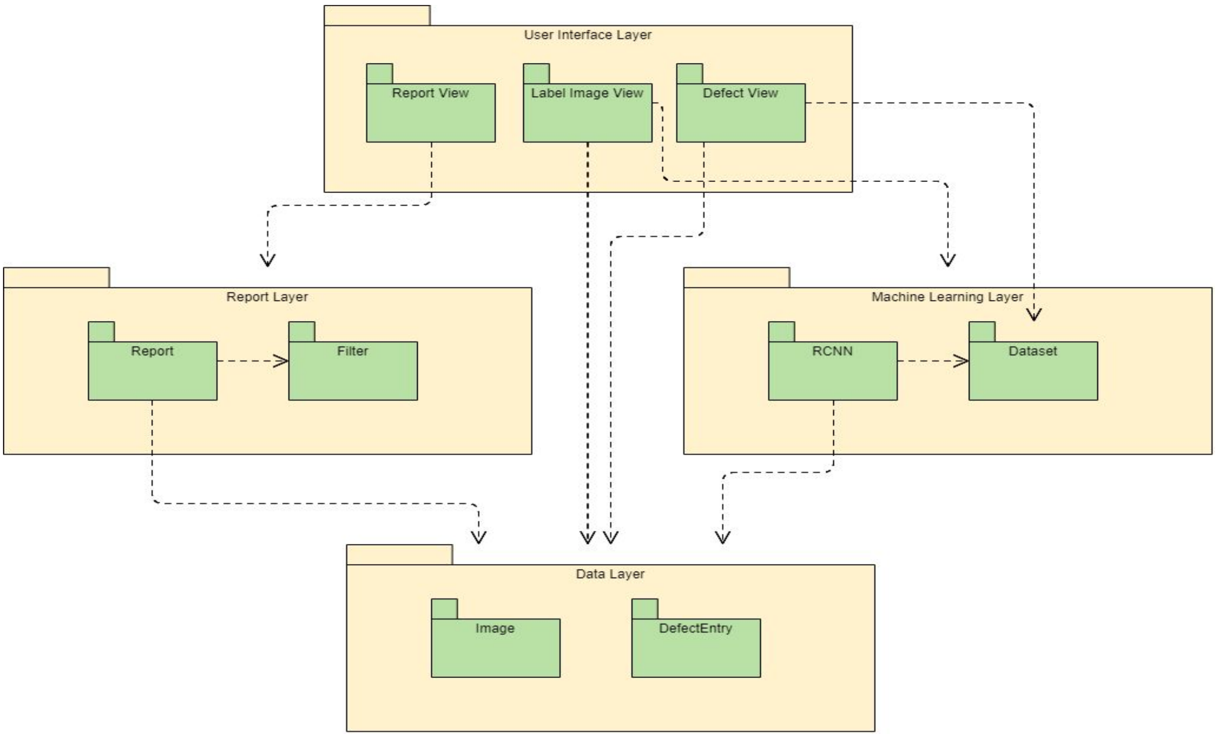

Figure 1: Subsystem Decomposition Diagram

3.3 Hardware/software mapping

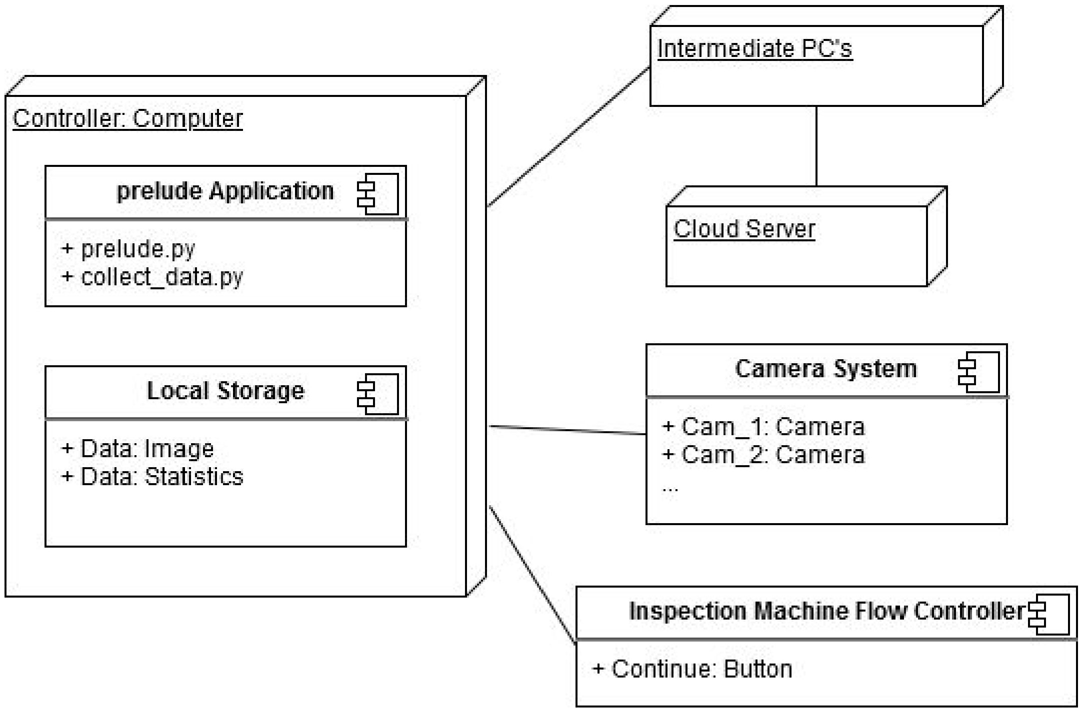

Figure 2: Hardware/Software Diagram

The four hardware components of our system will be the computer in the cloth inspection area in the company (named Controller), the camera system consisting of sufficient number of cameras for inspection (exact number will be determined after further field research) all connected to and to be controlled by the Controller, the cloud server for training on the data and intermediate PC's to manage the training.

In data collection phase, a worker(s) will use the computer for recording, labeling and saving of faulty images on the local storage via the application that we are going to implement (collect_data.py) after detecting them with the conventional method. One camera will be used at this phase and it will be connected to the Controller for this purpose. Data to be collected will be kept in the local storage of the Controller. For training, a cloud service will be used since preprocessing and actual training can require high computational load therefore need for high capacity GPU and RAM. Studies before training will be conducted via and training in the cloud service will be managed from our personal computers since direct connection from the Controller to Cloud Server may pose security problem. Once the necessary information (weights of the neural network) is extracted from the data in the cloud, they will be transferred from Intermediate PC's into the Controller in a secure way. In the prediction phase, multiple cameras are planned to be connected to the Controller, and the second application that we are going to implement (prelude.py) in the controller will be used to process the image stream coming from the cameras. Once a fault is detected, the Controller will stop the fabric flow in the corresponding inspection machine and display the identification related to the machine on the screen. After handling the fault, the worker will restart the fabric flow in that inspection machine by pressing on the “Continue” button near the machine. Statistical data related to the faults in the fabrics and handling process of these faults etc. will be stored in the local storage of the Controller.

3.4 Persistent data management

We will not use any databases in our software. We will persist the data using text and image files. Image files will be in any format (.png, .jpg, etc.) produced by the camera. In text file, we will hold class and bounding box for each image generated by our image labeling software. Since these .txt files and image files will be enough for our case, we do not want to complicate the project by incorporating databases.

At first, the data will be stored at the data center of the textile company due to security concerns raised by the company itself. Hence, the data will be stored locally in the company and be accumulated there. When the data count reaches some level, to train the model, this data will be reached from a dedicated or a cloud server where the algorithm will be trained. This data fetching process will be done securely not to breach security rules of the company.

3.5 Access control and security

During the data collection phase, the data will be collected and labeled by a worker via the application that we are going to implement and design, and will be shared with us regularly while we are working on the implementation of the program to be used as predictor. The regular sharing of the collected and labeled data requires some third party platform to be uploaded and downloaded. We are planning to use a reliable platform for this data transfer such as Google Drive since the physical transfering of the data from the company is not feasible due to long distance from the company. This issue was also discussed with the company and accepted as a secure solution. The application to be used during data collection and labelling and the data itself will not be allowed to be accessed by or shared with any third party unless such a requirement/choice emerges and the company’s approval is taken regarding to such an operation after the successful implementation of the solution. While training our model on the data collected, we are planning to use our personal computers, and from this point to the time the solution will be implemented in the company we will not share the implementation and the trained model with or load them to any platform/third party or load.

The second application to be used in the prediction phase will only be open to the workers' usage and the statistical data to be recorded in the prediction phase will only be open to responsible managers/engineers in the company. In this case, some accounts are planned to be created and given one to the each worker since the company management attaches importance to the work/worker recordings, and these type of recordings have already been kept, e.g. in fabric production stage. Additionally, some special personal or authorized account(s) with id and password can be created and the system can be designed in a way that it only allows the view of statistical data to such authorized workers. In case the statistical data are shared/sent to us for some further studies or update of the system, we will not share these statistical data with any third party and whether the transfer of this data will be done via the same method/platforms in the first data collection will be discussed and decided together with the company. Finally, the image and statistical data in the prediction phase will be stored in the local storage of the computer in the cloth inspection area and should only be accessible via the application for identification of the accessor.

3.6 Global software control

In our project, we will not be using any database or shared file. Therefore will not be facing any concurrency or synchronization problems.

3.7 Boundary conditions

Prelude will have three main boundary conditions for both data collection and prediction phase of our project which are initialization, termination and failure of the program. Each of these boundary conditions will be discussed in the following sections.

3.7.1 Initilization

For both data collection and prediction phase, in order to install the programs to the computers in textile company, an executable file will be sent to the manager of the company. After the camera is connected to the computer, this executable file will be started. Then our program will be able to save the pictures to the local storage of the computer. If the company needs to save the worker's id, we will be adding a login page at the beginning of our program.

3.7.2 Termination

For both data collection and prediction phase, workers in the textile company will be able to close the application without necessary interactions. If the company wants to collect any data of the id's of the workers who has used the application, they will need to log out from the program before closing the application.

3.7.3 Failure

The possible failures might be caused by several cases. These are power outage of computer, crashes because of some bug in our code or losing the connection with the camera. We will be saving all of these failures into a log file and take these files from the company manager in order to see why the failures occurred. Since we will be saving the necessary pictures to the local storage immediately after the picture is taken, we will not be losing any data. Additionally, for the prediction phase we will be sending 5V output to stop the machine. Since workers may not notice the failure and the machine might continue production without no one checking the textile quality.

4. Subsystem services

4.1 User Interface Layer

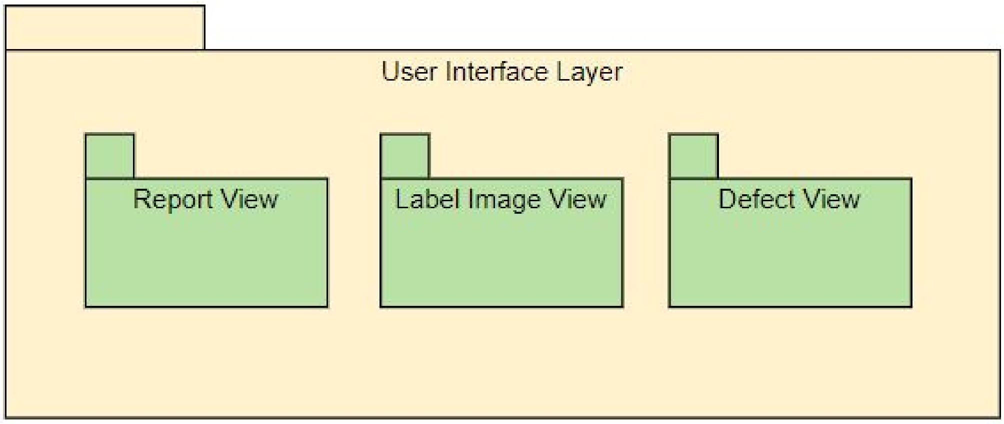

Figure 4: User Interface Layer

4.1.1 Report View Service

This service provides user interface elements to show detailed reports regarding the defects their frequencies and specifications of the clothes that defect occurred. It only uses services of Report layer.

4.1.2 Label Image View Service

This service populates user interface elements to provide image labeling functionality. User may draw rectangles using their mouses and may select the defects using comboxes.

4.1.3 Defect View Service

This service provides interface elements so that the user can see the lastly detected defects, its label and bounding box to physically locate the defect and fix it.

4.2 Report Layer

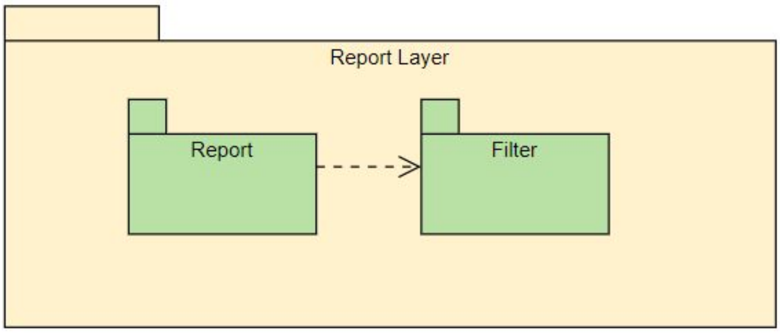

Figure 5: Report Layer

4.2.1 Report Service

This service is responsible from preparing a report about the system considering given filter. Reports include statistical information about the system such as total number of defects found in a period of time or most frequent types of defects.

4.2.2 Filter Service

This service provides a way to combine the user requests from a Report. The content that will be created by Report Service created considering the given Filter.

4.3 Machine Learning Layer

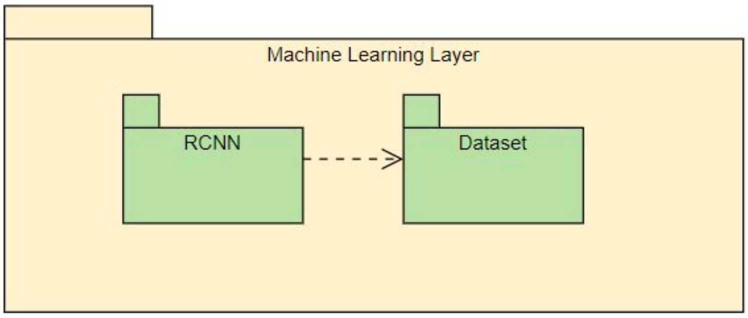

Figure 6: Machine Learning Layer

4.3.1 RCNN Service

This service constitutes the core of the system. It is responsible from both training and prediction phases. It specifies the network architecture. It uses the Dataset service to feed the network in both training and prediction phases.

4.3.2 Dataset Service

This service retrieves the data references from Data Layer services, loads and prepares the retrieved data to be used in RCNN Service and provides whenever requested.

4.4 Data Layer

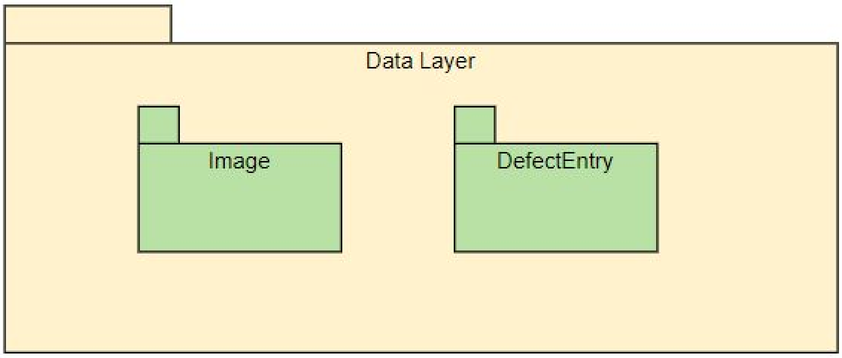

Figure 7: Data Layer

4.4.1 Image Service

This service stores the image references combined with image defects that are provided by DefectEntry Service if any.

4.4.2 DefectEntry Service

This service stores the information about the occurrence of defects such as their position and kind of fabric.

5. New Knowledge Acquired and Learning Strategies Used

So far we have learned two major concepts to help accomplish our project. First one is the camera choice for the project. We at first thought that a WebCam with 720p or 1080p resolution at 60 FPS would be enough for our purpose. To test our intuition, we requested from the people at the company to capture video at various resolutions and FPS values. Yet, the results were not satisfactory for us because even at 60 FPS and 1080p resolution images were blurry. We concluded that WebCam would not be feasible for our problem. We have searched for other types of cameras and especially looked at what other companies which work on computer vision use as camera. We have finally found that the popular type used in industry for these purposes is Line Scan and Area Scan cameras [5]. These both can capture high resolution pictures at industry standards. Among the two, Line Scan cameras are more widely used in capturing fast moving objects such as fabric because they can capture at quite high frequencies to end up with zero motion-blur [5]. Hence, we advised the company to purchase Line Scan cameras to capture the images. Secondly, we have learned R-CNN algorithms by getting our hands dirty on a real world problem. We picked a problem that can be solved by R-CNN algorithm which is to detect defects in a steel images. This was a rather easy task because unlike our project the training data was already captured and ready to be used. Hence, by trying to solve that problem we have learned the specifics of R-CNN algorithm and how it works. Throughout the learning process, we have also looked at Medium articles and the academic papers to get an insight about the algorithm. Hence, we now can apply our knowledge on this problem after getting the data.

6. References

[1] M. S. Biradar, B. G. Sheeparmatti, P. M. Patil and S. Ganapati Naik, "Patterned Fabric Defect Detection Using Regular Band and Distance Matching Function," 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA), Pune, 2017, pp. 1-6.

[2] K. Srinivasan, P. H. Dastor, P. Radhakrishnaihan, S. Jayaraman, "FDAS: A knowledge-based frame detection work for analysis of defects in woven textile structures", *J. Text. Inst.*, vol. 83, no. 3, pp. 431-44 7, 1992.

[3] CEO of Textile Company A, Interview, Nov 2019

[4] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

[5] Hanbay, K., Talu, M. F., & Özgüven, Ö. F. (2016). Fabric defect detection systems and methods—A systematic literature review. Optik, 127(24), 11960-11973.