2019 Trimester 2

**MTRN4230**

**Robotics**

**Group Assignment - Computer Vision (Decoration)**

Jonathan Loong z5014751

Sub-Report Workflow for Computer Vision - Decoration

**1. Preliminary Research**

**i. Qwirkle Block ML Research/Ideas**

The major task for the Computer Vision Engineer (Decoration), was to reliably execute object recognition on Qwirkle blocks. In order to achieve this, the development of a Qwirkle O**bject Detection** program was required. This can be broken up into two parts, requiring ‘image classification’ and ‘object localisation.’ In the context of the CAKE Proof of Concept:

* **Image Classification** – prediction of which Qwirkle block are needed and assigning a class ‘labels’ to them (eg: Criss cross, circle).
* **Object Localisation** – the customer image will contain an arrangement of blocks. The coordinates of the centre of each block is to be identified (and sent to robot arm)
* **Object Detection** – combination of both in real time (on conveyor)

Using MATLAB, this program can be developed in a variety of ways:

1. Transfer Learning. Fine-tuning a previously trained neural network and adjusting it for the Qwirkle training database.

Advantages:

* Faster to train

Disadvantages:

* If using existing networks like googleNet, the Qwirkle blocks may not work well

2. R-CNN. Region-based Convolutional Neural Network. Locates many regions of interest for object classification.

Advantages:

* Allows for a selective search of objects of interest
* Relatively simple method for object localisation and classification

Disadvantages:

* Slow. Generates around 2000 proposed regions in any input image

3. YOLO (You Only Look Once). Predicts and labels bounding boxes in a single evaluation. Simple structure in a single network.

Advantages:

* Possible to use in real time (up to 45 fps)
* Pretrained using the imageNet dataset

Disadvantages:

* Multiple/overlapping bounding boxes are a possibility
* Lower accuracy in relation to object localisation (might be issue with final decoration placement). YoloV2 may have some better performance/accuracy

**ii. Proposed ML Algorithm**

Based on prior research and the capabilities of the MATLAB Deep Learning Toolbox, it was decided to develop the Qwirkle Object Detection program using Faster RCNN.

**iii. Outline Decoration Detection Pipeline**

Here is a flowchart to present the development of a Qwirkle object detection program using both the cameras in the robot cell and the conveyor camera. In particular, the interaction/relationships between the two cameras and the rest of the robot cell was considered (see Figure 1).

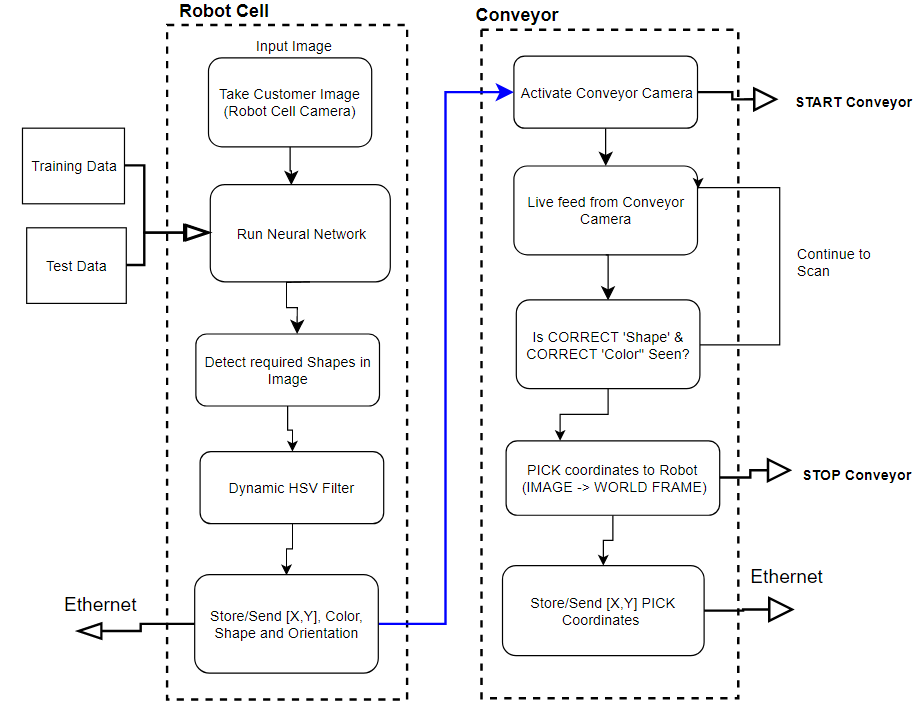


Figure 1: ML/CV Pipeline

**2. Calibration of Robot Cell and Conveyor Camera**

**Intrinsic**

The intrinsic parameters for the robot cell camera are concerned with the focal length and distortion properties of the camera. Calibration required performing a projective transformation to relate the camera frame to the image plane, as in Figure 2 and Figure 3.

1. Measure width of one square side of the checkerboard pattern (length = 25mm x 25mm )
2. Obtain 10-20 images at the distance between camera and OOI (height from robot camera to table top)
3. Place checker board at angles less than 45 degrees relative to camera plane
4. Different orientations relative to the camera
5. Using Single Camera Calibration Toolbox in MATLAB

After the initial round of calibration, the average re-projection error was found to be 0.11 pixels. This error is the distance in pixels for which the detected points differs from the re-projected points. After the removal of the two outliers, the error was reduced to 0.09 pixels.

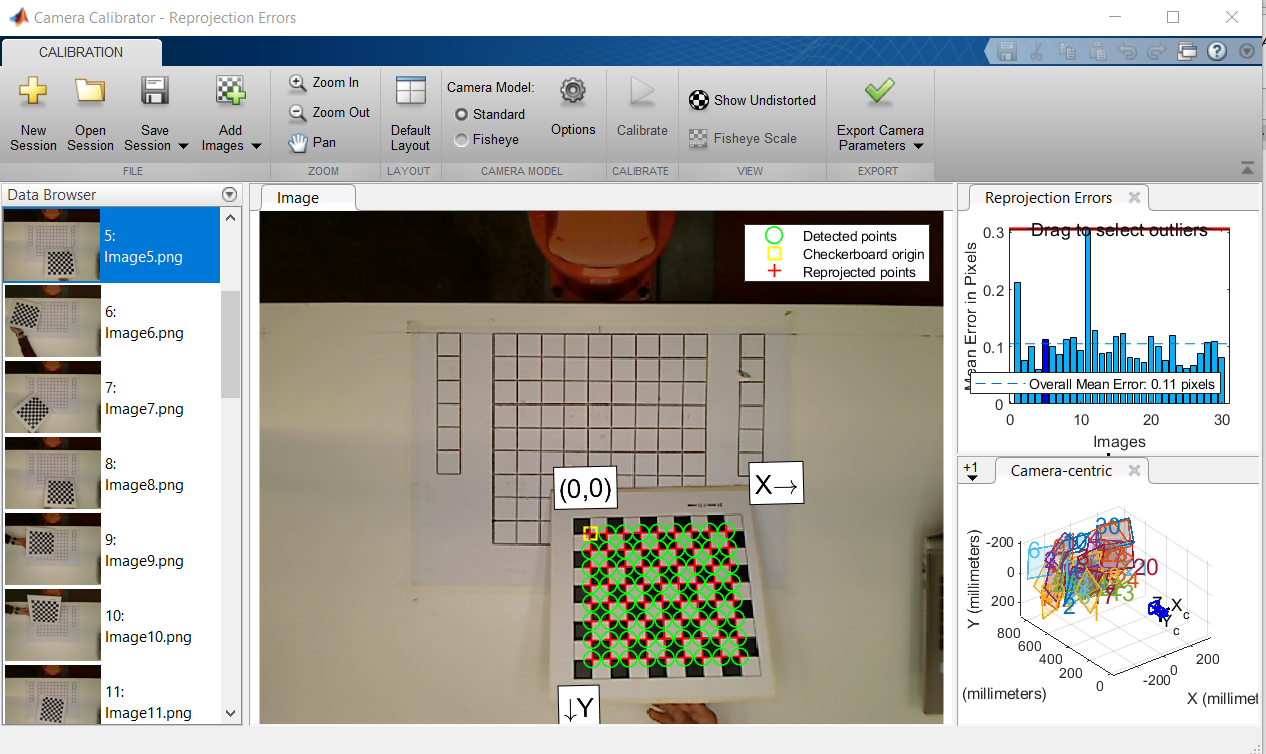


Figure 2: Initial Intrinsic Calibration

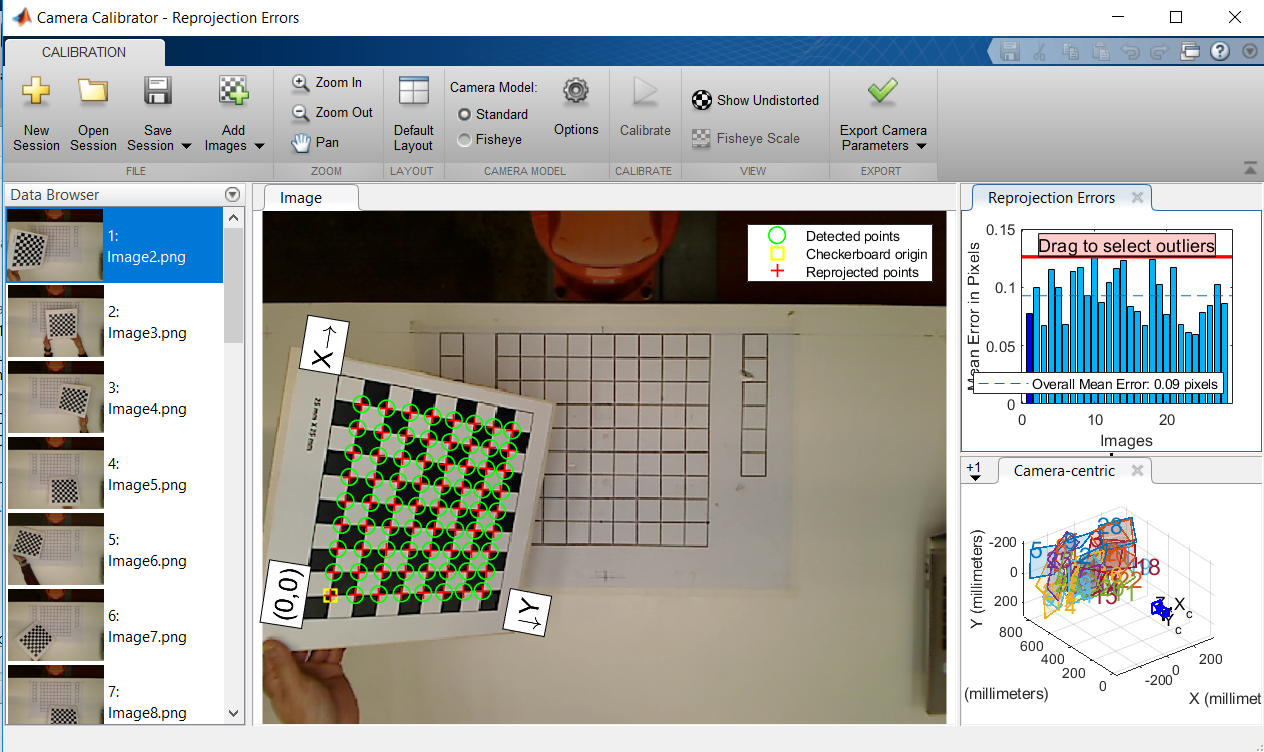


Figure 3: Final Intrinsic Calibration

The intrinsic matrix K (Calibration matrix) was found to be:

Kint =

**Extrinsic**

Using the following points as control reference points, the pixel values for each (u,v) were estimated using the following 3D points noted as (Xw,Yw,1) as listed in Table 1:

Table 1: World and Camera Coordinates for Ref Points

|  |  |
| --- | --- |
| **3D Point (World Frame)** | **2D Point (Camera Frame)** |
| T2 (175, -520, 149) | (7,289) |
| T1 (175, 0, 149) | (800,289) |
| T3 (175, 520, 149) | (1597,289) |
| T4 (548.6, 0, 149) | (800,855) |
| T5 (548.6,520,1409) | (1597,855) |

The correspondence (homography) between these reference points in the image plane (u,v) and the world coordinate frame (X,Y,Z) enables the determination of the extrinsic matrix for the robot cell camera.

Translation vector – transformation from world coordinate to camera coordinate

TranslationVector =

Rotation vector – position of camera in world coordinate frame

RotationMatrix =

**3. Creating Qwirkle Detector Network**

A combination of ‘Ink Writing’ (Letters) and ‘Chocolate Blocks’ (Colored Qwirkle blocks) make up the customer’s sample pattern. Focus for the decoration part was on the reliable detection of the correct Qwirkle block and the required position and orientation.

**3a. Training Data**

To create the Qwirkle block detector, training images were required for Machine Learning. These blocks will be required to be successfully detected at both the robot cell and the conveyor. Using the ‘Image Labeler’ App, a ROI (bounding box Region of Interest) was created for the variety of blocks. These created the database for the ‘ground truth’ images to be passed into the Machine Learning network (see Figure 4).

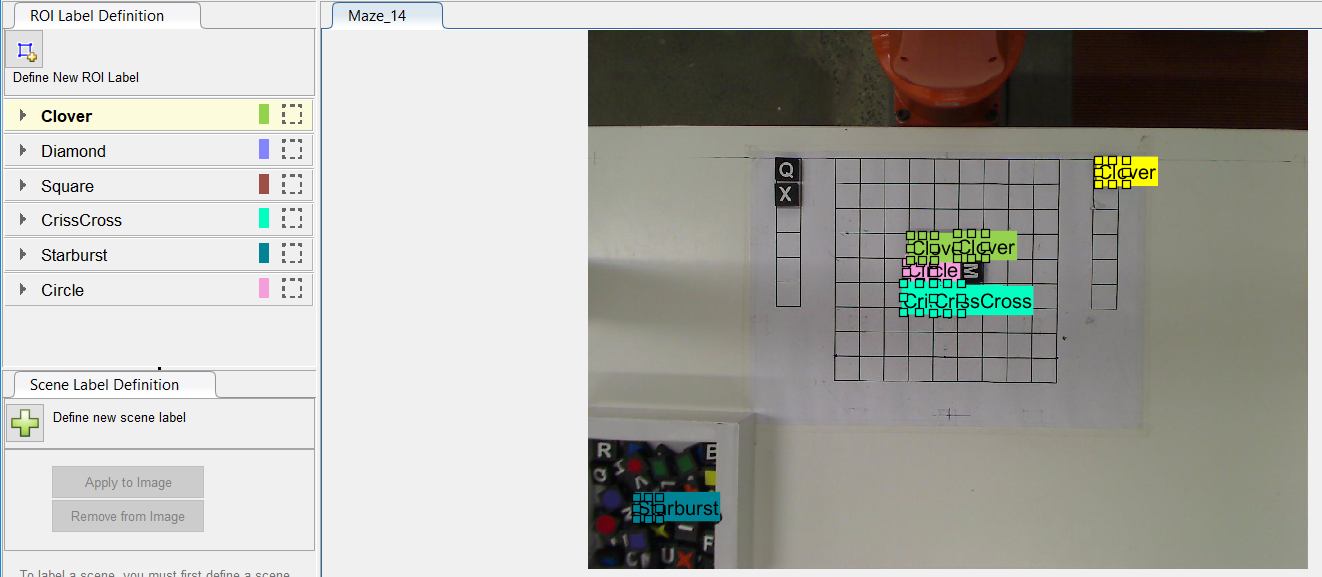


Figure 4: Labelling Ground Truth Images

**3b. Creating the ML Network**

As detailed in Section 1, a Faster R-CNN detector was tested for the prototype cake decoration system as outlined in Figure 5:

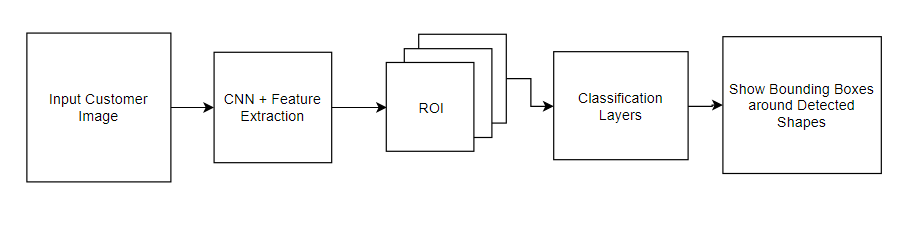


Figure 5: Faster R-CNN Flowchart

The Faster R-CNN detector used bounding boxes to identify which Qwirkle block was found in the image. For each bounding box, the following was predicted:

* Dimensions of the bounding box
* Confidence of having an object in the box
* Probabilities of which class the object belongs to

The Faster R-CNN detector used a pre-trained Resnet-50 network for the feature extraction. The first sub-network following the feature extraction network was a region proposal network (RPN) which provides options for objects. The second sub-network is trained to predict the actual class of each proposal (the different Qwirkle shapes).

**i) Making Training Dataset**

Labelled ‘ground truth’ images of Qwirkle blocks taken from the Conveyor and Robot Cell cameras at 1600 x 1200 resolution (for consistency) but resized to 800 x 600 for input into the neural network. See Figure 6 and Figure 7.

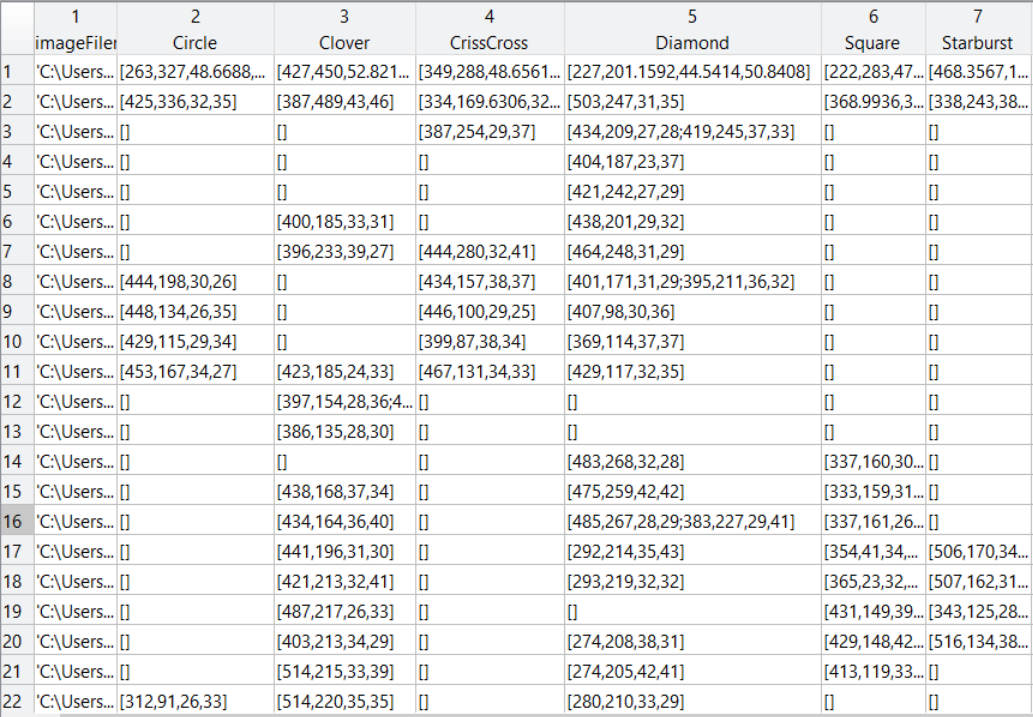


Figure 6: Training the ML Network – Bounding Boxes

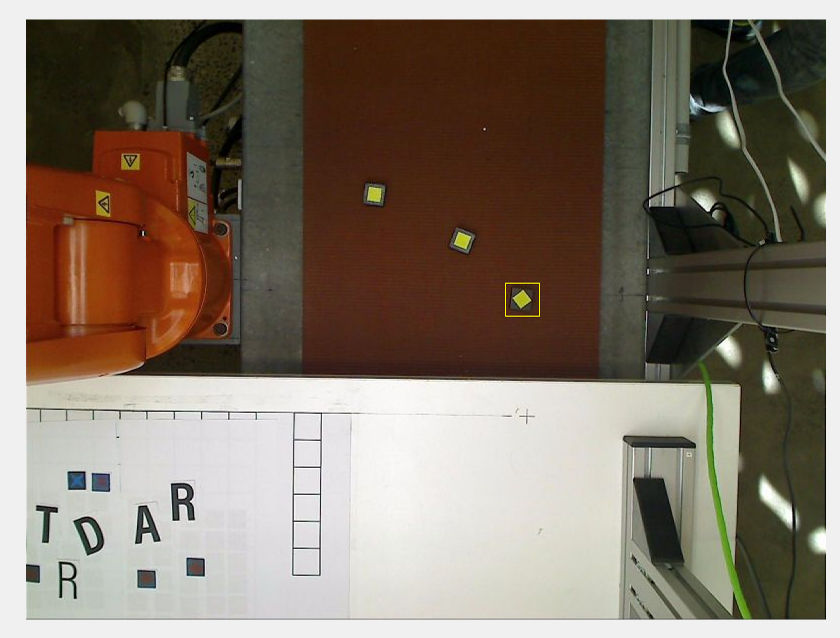


Figure 7: Training the ML Network – Sample Training Image

Some of the important training parameters are outlined in Table 2 and Figure 8 presents a screenshot of the initial training process.

Table 2: Training Parameters

|  |  |
| --- | --- |
| Epochs | 10 |
| Batch Size | 1 |
| Learning Rate | 0.001 |
| Function | Stochastic Gradient Descent with momentum |

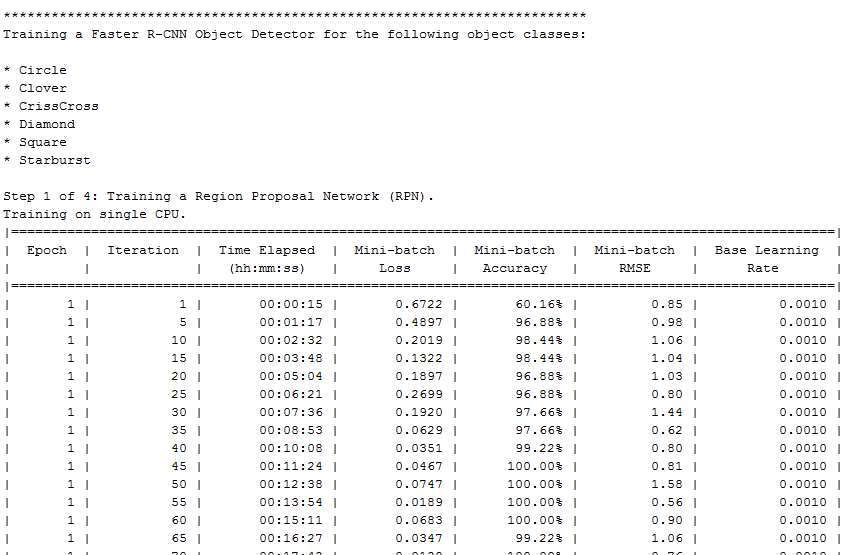


Figure 8: Screenshot of Initial Training Faster R-CNN

**ii) Training Loss**

The purpose of the Qwirkle Detector is to reliably and quickly detect and classify which classes/patterns of Qwirkle blocks are present in the current image frame. The training loss refers to the accuracy with block classification for each mini batch. There is the danger of having over-fitted the model, where there was low loss during training but inaccuracy when predicting new data. (ie: new blocks on the Conveyor). See Figure 9, which demonstrates how between Step 1 and Step 4 of training the Faster RCNN network, the general loss reduced to below 0.1 which was desirable (after training for total of 10 epochs).

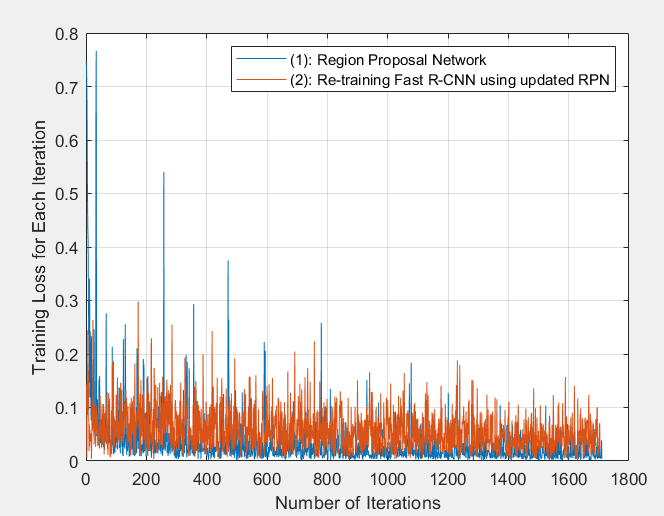


Figure 9: Graph of Training Loss

**iii) Evaluating the Qwirkle Detector**

To test the accuracy of this trained network, sample test images (not from training set) were loaded into the workspace and the prediction score and location of the bounding box was observed. As noted in Figure 10, the following Qwirkle blocks were detected quite successfully, taking into consideration the limited training time and training dataset.

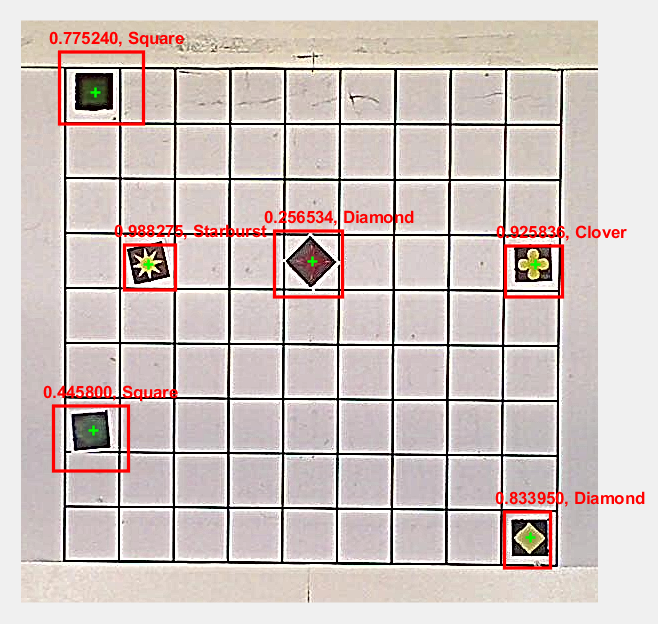


Figure 10: Result of Network on Test Image

The first metric was precision, the number that incorporates the ability of the detector to make correct classifications (precision) and the ability of the detector to find all relevant objects (recall). See Figure 11:

Figure 11: Graph of Network Precision over Time

**4. Customer Image Computer Vision Pipeline**

After the object detector, the Qwirkle Detector has implemented Image Localisation and Image Classification. The next stage in the image processing pipeline was to determine the matching ‘Shape’ and ‘Color’ for each Qwirkle block.

**4a Localisation and Color Classification of Blocks**

After the object detector was run, the detected shapes and their estimated position was obtained. A dynamic HSV filter was implemented to go through each color of RGBY (1234) and locate the largest blobs for the each color. A threshold pixel area size was set to filter out unnecessarily small blob detections. The array of centroids for each color were then sorted in descending order. The largest centroids were analysed by area and suitable blocks were stored into an array, storing location (image coordinates) and encoded color.

Since the object detector only was able to recognise the shape (eg: clover vs starburst), the correct color had to be matched as well. By correlating the centroid of the blob to the bounding box geometric centre for each space, the required shape in the right color was determined.

**4b. Determining Block Pose**

The next step was deriving the orientation for the required blocks. The SURF detector was applied to

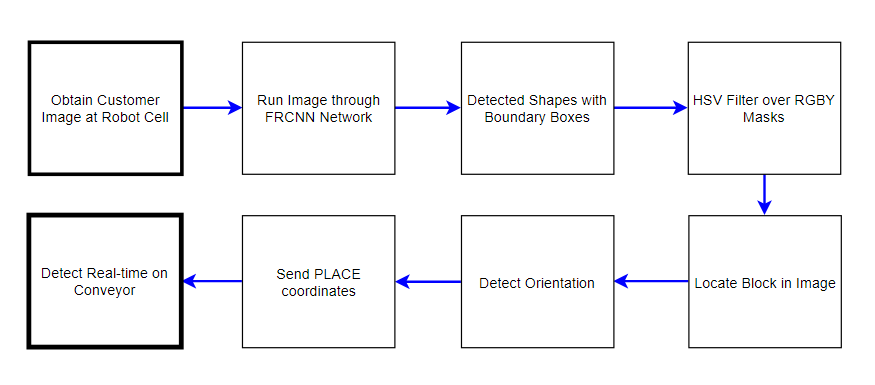
**5. Detecting Blocks on Conveyor**

Dynamic object recognition in real-time for blocks incoming from conveyor. Faster R-CNN was capable of detecting objects in a reasonable period of time.

**6. Sending Info to Robot Arm**

The computer vision pipeline was implemented and utilized in two key areas of the Cake Decoration prototype system. In particular, the program needed to send the world coordinates of blocks (PICK and PLACE coordinates) and the orientation of blocks to the ARB 120.

In summary, Figure 12 presents a diagram showing the image processing pipeline for the decoration detection system.



­Figure 12: Summary Diagram of Computer Vision Pipeline