Information Extraction from Hand-marked Industrial Inspection Sheets

Gaurav Gupta, Swati, Monika Sharma, Lovekesh Vig
TCS Research, Delhi
Tata Consultancy Services Limited, Gurgaon - 122003, INDIA
Email: { g.gupta7 | j.swati | monika.sharma1 | lovekesh.vig }@tcs.com

***Abstract*—In order to effectively detect faults and maintain heavy machines, a standard practice in several organizations is to conduct regular manual inspections. The procedure for conducting such inspections requires marking of the damaged components on a standardized inspection sheet which is then camera scanned. These sheets are marked for different faults in corresponding machine zones using hand-drawn arrows and text. As a result, the reading environment is highly unstructured and requires a domain expert while extracting the manually marked information. In this paper, we propose a novel pipeline to build an information extraction system for such machine inspection sheets, utilizing state-of-the-art deep learning and computer vision techniques. The pipeline proceeds in the following stages: (1) localization of different zones of the machine, arrows and text using a combination of template matching, deep learning and connected components, and (2) mapping the machine zone to the corresponding arrow head and the text segment to the arrow tail, followed by pairing them to get the correct damage code for each zone. Experiments were performed on a dataset collected from an anonymous real world manufacturing unit. Results demonstrate the efficacy of the proposed approach and we also report the accuracy for each step in the pipeline.**

I. Introduction

Factories, power plants, airlines and other industries that rely on heavy machinery need to routinely conduct inspections to ensure that their machines are in good health and assess for any obvious wear and tear or damage to the different components of the machine. This inspection is often conducted in the field manually and the results are noted on paper based inspection sheets having line drawings of the different zones of the machine. Inspection engineers are hired to note the assessment of each particular zone of all the machines and thereafter, record them on the inspection sheet. These inspections sheets are then camera scanned and archived for future reference. Because of this reason, inspecting various zones of machines is a laborious and expensive process. Thus, as more and more of these industries get digitized, there is a need for either augmented reality based inspection [1] or to extract the hand-written information corresponding to the relevant zone from these inspection sheets for ready access. These inspection sheets have a particular commonly used format and the inspection engineers are trained to note down their assesments according to a provided code for different types of faults such as cracks, leaks etc. The code is written on the inspection sheet and connected to the concerned part in the machine diagram via a hand-drawn arrow as shown in Figure 1. Several organizations, over the years have conducted a large number of such inspections which have resulted in the generation of millions of inspection sheets from which information is to be extracted and digitized. In this paper, we propose a computer vision and deep learning based pipeline to extract information from such inspection sheets with high accuracy from an anonymous real world dataset. The pipeline as shown in Figure 3 proceeds in two stages: (1) *Specific Component Localization* - It involves using template matching, for accurate part localization from the standardized machine diagram, localization of arrow heads and tails using a combination of Faster-RCNN [2] and a regression model using a deep convolutional neural network (deep CNN) inspired from Sun et. al [3]. Simultaneously, we use connected components [4] to identify the different handwritten text segments. (2) *Zone and Text Assocation* - It involves mapping the machine zone to corresponding text segment in the sheet using the arrow head and tail information obtained in stage 1.

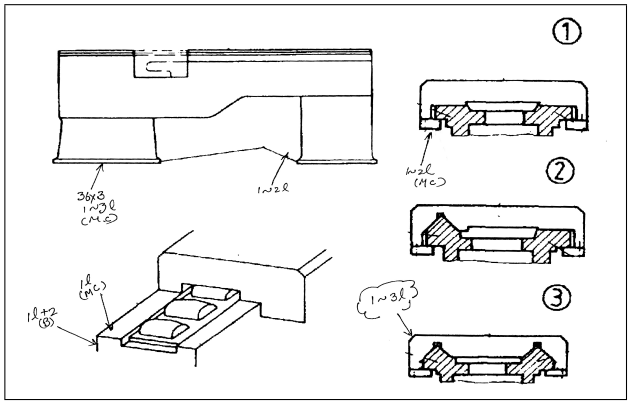


Fig. 1. A similar example of industrial inspection sheet consisting of line diagrams for various zones of machines, and their damaged parts indicated via handwritten codes and arrows.

This paper makes the following contributions:

- We propose the localization of individual zones of standard machine diagrams using template matching, localization of arrow heads and tails using a combination of Faster-RCNN [2] and convolutional regression network [3] as discussed in Section III-A1 and III-A2 respectively.
- We propose a method to map machine zones with hand-

written text segments using arrow head and tail information as detailed in Section III-B
- We conduct experiments on a dataset of real world inspection sheets and evaluate the efficiency of our proposed method while reporting the accuracy at each stage of the pipeline and the accuracy of individual components of the pipeline. Results are encouraging as shown in Section IV.

The rest of the paper is organized as follows: Section II discusses related work in the literature. Section III outlines the steps involved in the method used for each component in pipeline, which contains Section III-A explaining individual part localization and Section III-B explaining text mapping and zone mapping with the arrows. Section IV details the experimental setup and presents the results. Finally, Section V concludes the paper and presents potential avenues for future work.

II. Related Work

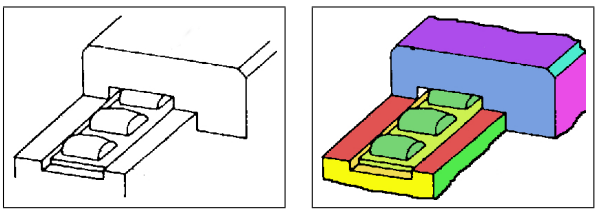
While plenty of research studies are available on document analysis and text reading [5], [6], there exists limited work that address the challenges stemming from documents such as machine inspection sheets. The main objective of the paper is to automate the task of reading inspector's comments from hand marked inspection sheets of various machines in power plants, factories etc. Although numerous attempts have been made in the literature on different components of our proposed pipeline but an end-to-end complete solution for information extraction from handmarked inspection sheets does not exist. We have used a template matching technique to find different machine zones in the line diagram of inspection sheet. Template matching [7], [8], [9] is a primary technique in computer vision for object detection which attempts to find a sub-image from a target image after matching with a reference image called template. We observe that arrows are useful fiducial markers present in the inspection sheets as shown in Figure 1, to associate machine zones with corresponding text. Conventional methods for arrow detection are based on geometry based features [10], edge maps of arrows processed using hough-transform [11] and multi-class support vector machines for arrow recognition from image maps extracted via projection histogram on Inverse perspective image [12]. We trained a deep learning based object recognition algorithm Faster-RCNN [2] for detecting arrows in the inspection sheets followed by deep CNN based regression model which inherently learns the handwritten arrow structure. Faster-RCNN is a deep learning based model for component localization which consists of a Region Proposal Network for generating region-proposals and Region-based Convolutional Neural Network (RCNN) [13] for object detection. Text localization is a well studied problem in the machine learning community which basically involves localizing text-regions in the image and subsequently, reading the corresponding text [14], [4], [15]. Given that the foreground is well separated, connected component analysis for detecting text in real world images [4] and container code recognition on shipping containers [16] has proven to be effective. Therefore, we use connected components analysis to localize text regions using the arrow head and tail information for cues.

III. Proposed Methodology

In this section, we describe our proposed pipeline for information extraction from industrial inspection sheets as outlined in Figure 3. The pipeline contains several modules which involve detection and localization of salient objects like machine zones, arrows and hand-written text-segments followed by mapping of the text-segments with the corresponding machine zones using arrow head and tail position. The mapping is one-to-one, in a way that each machine is mapped to only one text segment and vice-versa, where the arrow behaves like a linker for mapping. The proposed method, in effect, is divided into 2 major sections: Section III-A explains Individual Part Localization which gives us the contour of each zone of interest as an array of pixel locations in the machine line diagram, pair of head and tail position of each detected arrow and bounding boxes around text-segments present in the sheets as shown in Figure 1. The second section III-B is Mapping, explaining the method for one-to-one mapping of machine zones with the corresponding text segment using arrow head and tail positions.

A. Specific Component Localization

1) Machine zone localization: The engineer performing inspection of different machine zones for any defects, writes codes and comments manually against every machine zone in the inspection sheet. Hence, we begin by localization of every machine zone of the line diagram in the inspection sheet. As the printed inspection sheets have a standard template and are scanned by camera with same height, orientation and intrinsic camera parameter, the structure and size of all line diagrams of machines in each sheet are the same. Although their relative location in sheets do vary slightly. Therefore, we use template matching to localize the machine components in every inspection sheet. We have created reference templates, T_k for $k \in \{1, N\}$, indicating different N machine templates for each line diagram (one example is shown in Figure 2) and stored the locations of different zones as set of contours (C_k) in the database.



(a) **(b)**

Fig. 2. (a) Reference template of a machine diagram, and (b) Various machine zones marked with different colors.

While the reference template is slided over rows and columns of inspection sheets one pixel at a time, it calculates a cross-correlation metric and then maximizes it over all rows and columns. Assuming the highest correlation point (l_k) as

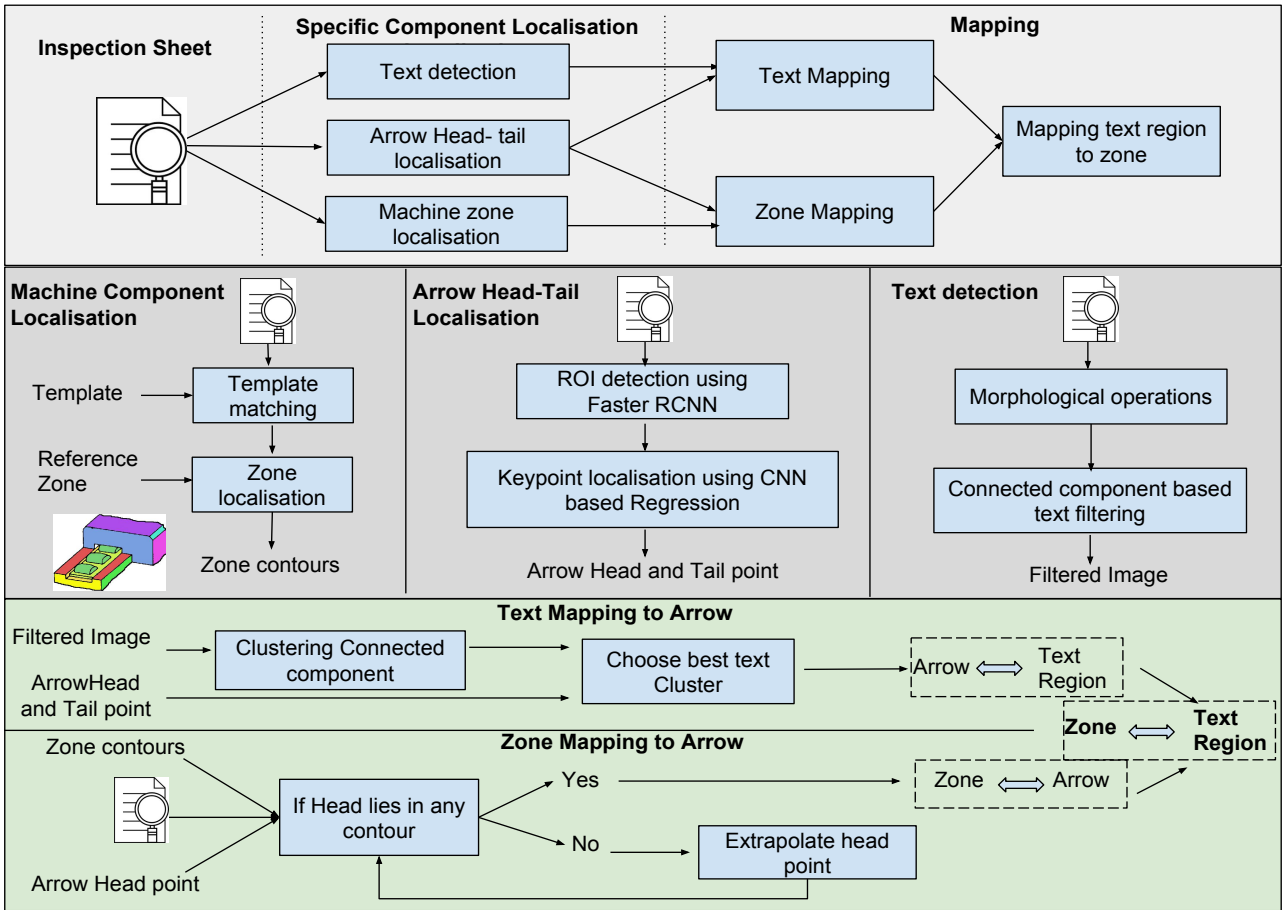


Fig. 3. Flowchart depicting the proposed methodology for extracting information from inspection sheets. The first stage involves independently localizing the arrows, the text regions and the machine zones. The second stage involves mapping arrow heads to machine zones and arrow tails to text regions. The machine zones can then be directly mapped to the corresponding text regions.

the top-left coordinate of a machine diagram in the given test inspection sheet, we find contours of corresponding machine zones (Z_k) by making use of l_k according to the relation given in Equation 1.

$$Z_k = C_k + l_k \tag{1}$$

2) Arrow Head and Tail Localization: Another important component in the inspection sheets is the *arrow*. Arrows serve as the connecting entity which are used to map the text segments with the corresponding machine zone. The handwritten text is present at the arrow tail and the corresponding zone of the machine line diagrams is present at the arrow head. The inspection sheets contain handwritten arrows which make the detection task difficult using traditional mathematical models. Hence, we are utilizing deep neural network models to learn the arrow structure. This is performed in two steps: first, we find the Region of Interest (ROI) and then we localize the arrow head and tail points. ROI consists of a rectangular boundary around the arrows in the inspection sheet. We have trained a Faster-RCNN [2] model on a set of inspection sheets to detect all the arrows. We observed that there exists situations where two or more arrows lie just next to each other, resulting in multiple arrows in the same ROI. This scenario creates confusion for arrow head and tail localization. To circumvent this problem, we trained a Faster-RCNN on partial arrows. Partial arrows imply arrows with arrow heads and a part of arrow shaft attached to head as shown in Figure 5(a). We have used the trained Faster-RCNN model to find ROIs in a given test inspection sheet. Further, we use the resultant ROIs to localize arrow head and shaft end (i.e. arrow tail). We have used a deep convolution neural network (CNN) based regression [3] model whose architecture is shown in Figure 4, to detect arrow head and tail key-locations. The details of the model are given in Section IV. The CNN based regression model predicts the pixel locations of the arrow head and tail in the given sheet, from which we can derive the direction in which the arrow is pointing. Some of the detected arrow tail points are not exactly matched with actual tail points, but are good enough for text mapping as explained in further sections.

3) Text Localization: In this step, we first remove noise like unwanted and spurious text, bubbles etc. from the inspection sheets. Due to the standardized format of inspection sheets, a lot of things are irrelevant and repeated objects exist in the sheets. Therefore, we remove this unwanted information from every inspection sheet using background subtraction. It

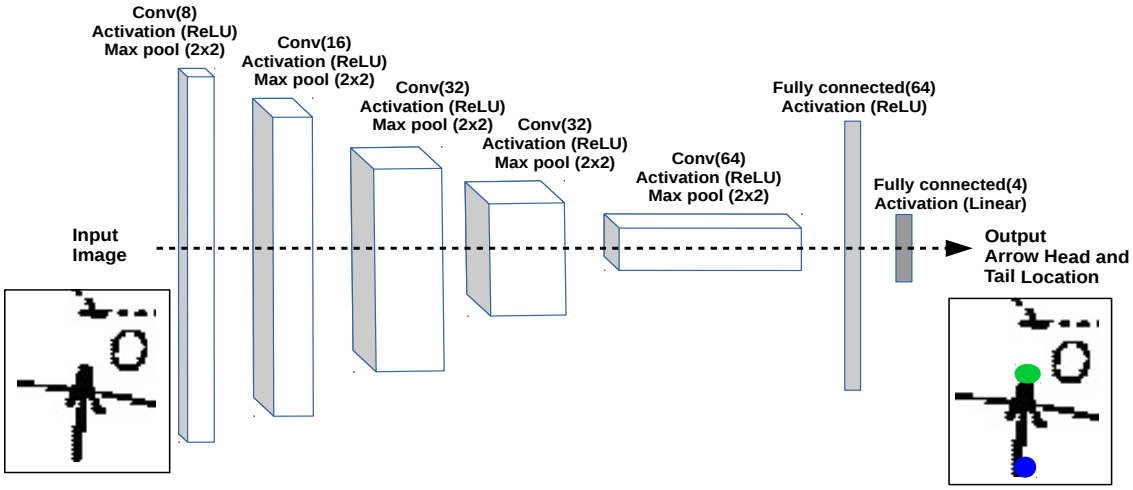
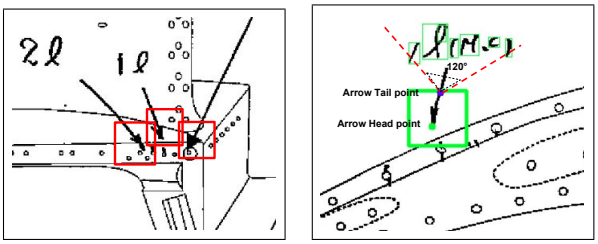


Fig. 4. Architecture of deep CNN based regression model used for localization of arrow head and tail. The input patch is the ROI extracted from the inspection sheet by arrow detection method. Output is the detected keypoints for arrow head(green color) and tail(blue color).

is carried out using template matching, where the reference template is subtracted from the test inspection sheet. Further, we use morphological operation of median filter to remove noise of tiny residuals. Once we have a pre-processed image, we use connected components analysis to get bounding boxes for all text segments and objects present in the inspection sheet. Overlapping regions are removed using Non-Maximum Suppression (NMS). Next, we apply empirically chosen upper and lower threshold of 4000 and 100 square pixels, on the area of these bounding boxes to get relevant text segments from the inspection sheet. The thresholds are empirically defined and are based on the image resolution. We may still get some undesirable patterns after this step because of the presence of a variety of unavoidable patterns. Hence, we use further processing to get superior text segments during text mapping as explained in Section III-B1.

B. Mapping

1) Text Mapping: To remove unwanted patterns in the obtained text-proposals from Section III-A3, we use the arrow tail points and the direction of arrows found in Section III-A2 and find the best Region of Interest for text (ROI-Text). It is observed from inspection sheets that the text is most likely to be present in the vicinity of arrow tail. Based upon this observation, we filter out the text-regions which do not lie in the vicinity of the arrow tail. We achieve this by using euclidean distance based thresholding. The text proposals should lie within empirically determined upper threshold (th_u) of the distance from the arrow tail, where $th_u = 150$. Subsequently, we apply flat clustering [17] with a cluster threshold of 1.2, on the text proposals. We choose the cluster with its mean in the opposite direction of arrow and closest to the arrow tail. To achieve this, a sector of 120^o on arrow tail, symmetrically around line of arrow, is taken as allowed region for cluster mean point (Refer Figure 5(b)). If we get multiple clusters in this sector, the closest one from the tail is chosen, which is considered to be the relevant text-region for the detected arrow.



(a) (b)

Fig. 5. (a) Bounding box annotations for training Faster-RCNN on partial arrows which contains arrow head and (b) Text-mapping using arrow-tail and text-segments.

2) Zone mapping: For mapping of zone contours discovered in Section III-A1 with corresponding text-regions obtained from Section III-A3 in the inspection sheets, we use the arrow head and direction to check if an arrow head occurs inside any zone contour. We have used the *Ray Casting Algorithm* [18] to find if the point lie outside or inside the polygon contour. We assign the head point to the zone contour which lie inside or touches its boundary. In cases where the head point is not found to lie inside or at the boundary of any contour, we extrapolate the head point in the arrow-pointing direction until it lies inside some zone contour. If the arrow head is at (x_h, y_h), unit vector in arrow head direction is (u, v) then next extrapolated point(x_1, y_1) is given as:

$$(x_1, y_1) = (x_h, y_h) + \alpha(u, v) \tag{2}$$

where α is the step size. If size of α is small, more steps are needed to reach inside a particular zone contour and if it is large, then it may not be able to find the zone contour. So, we have chosen a value of α = 30 which is the average distance between center and boundary point of minimum area zone contour in the sheet, which is logically an optimal step size. Also in some cases, where the extrapolation takes more than $3-4$ steps, the arrow is not exactly pointing in the direction of the zone but its head is close to the boundary. In such cases, the nearest zone from the head is mapped.

Using the above mentioned approaches for text and zone mapping, the overall mapping of text to zone is performed using the common arrow as a linker. It will be a one to one mapping that addresses the objective of this paper.

IV. Experimental Results and Discussion

A. Dataset

We have implemented our end to end pipeline on a dataset of anonymized inspection sheets provided by a company employing heavy machines. We have a total of 330 camera scans of these inspection sheets and they are divided randomly into disjoint set of 280 and 50 sheets. We have used 280 for training, finding templates, optimizing parameters and training the end to end pipeline. The other set of 50 inspection sheets is used for testing our approach. All sheets are scanned using camera with same position, orientation and intrinsic parameters. The resolution for each scan is 3210 x 2200.

B. Results and Discussion

The inspection sheet represents 8 different kinds of machine structures. All these structures contain a total of 86 subparts constituting different zones. Hence, we have 8 different reference templates. These are taken from a random image in the train set. We achieved perfect template matching on the complete test set and thus, our entire pipeline is error free with respect to template matching.

The Faster-RCNN [2] is trained on the manually annotated arrow images from the complete training set using the Zeiler-Fergus network with random weight initialization. We have trained this network for $20000, 10000, 20000$ and 10000 epochs, respectively for Stage1 RPN, Stage1 RCNN, Stage2 RPN and Stage2 RCNN. Rest of the training settings are taken as default as given in [2]. The accuracy is calculated as the percentage of correctly obtained ROIs out of the total arrows present in the test set. By keeping the confidence threshold greater than 0.9 and Non Maximal Suppression (NMS) threshold less than 0.05, our model is able to detect 171 correct ROIs out of 179 and 3 of the detections were obtained as false positives. The ROI detection is assumed to be correct if it fully contains the arrow head. The accuracy obtained for Faster-RCNN is **95.5%** (Refer Table I). The accuracy obtained is significantly high and the 8 arrow regions which are failed to be detected are among the ones having closely drawn arrows, and hence affected by the strict NMS kept for maintaining minimal false positive rate. Some examples of detected ROIs are shown in Figure 6.

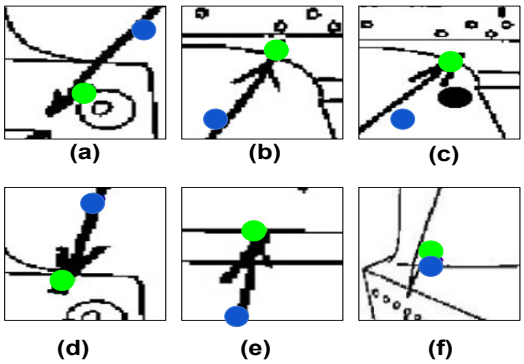


Fig. 6. Results for arrow localization on ROI given by Faster-RCNN. Green dot represents arrow head point and blue dot represents arrow tail point. The detected point does not lie on the arrow always as can be seen in (c) and (f)

The cropped images of partial arrows, taken from the train set is used to train our Deep CNN regression model. There are total of 1000 arrow images, which are divided randomly into 800 and 200 sets for training and validation, respectively. The model comprises of 5 convolution layers with $8, 16, 32, 32$ and 64 filters respectively, followed by 2 fully connected layers. Each layer except the last fully connected layer uses Rectified Linear Units (ReLU) as their activation function. Each convolution layer is followed by a Max-pool layer of size 2×2. Each convolution layer uses 3×3 kernel sized filters. The last fully connected layer has 4 hidden units representing x and y location of the arrow head and tail. It uses a linear activation function. We have used the *Adam* optimizer with default hyper-parameters to optimize mean square error cost function.

The number of epochs used in training is 500, where we achieve the highest validation accuracy. The input size of our images is 150×150. During testing, we obtained a mean square error of **170.3** for a set of 171 ROI images obtained from Faster RCNN. It implies a circle of radius of approximately 13 pixels in the image plane where the expected outcome would lie. If manually annotated ROIs on test set are given, the network gives mean square error of **148.1** for a set of 179 ROI images. It depicts the absolute error measure of our Deep CNN regression model.

The output from Section III-A2 is used for text detection. We measure the accuracy of the detected text box by finding *Intersection of Union (IoU)* between annotated text box and obtained text box. We choose IoU threshold to be 0.9. Using this, we are able to extract 157 correct text boxes at arrow tail out of 171 detected arrows (ROIs) by Faster RCNN. This provides us an accuracy of **91.8%**. With the manually annotated ROIs and arrow head and tail points on test set, we are able to extract 166 correct text boxes at the arrow tail out of 179 arrows. This yields an accuracy of **92.7%**, which is the absolute error measure of text detection (Refer fourth row of Table I).

Next, one-to-one mapping from arrows to the machine zones is performed. We are able to map 162 arrows correctly to

their corresponding zones out of the 171 detected arrows, thereby obtaining an accuracy of **94.7%**. The accuracy of zone mapping depends largely on the accuracy of head and tail point localization. With manually annotated ROIs and arrow head - tail points on the test set, we are able to map 178 arrows correctly to their corresponding zone out of the set of 179 arrows. Hence according to the absolute error measure, it is **99.4%** accurate as given in Table I.

It should be noted that the error at each step of the pipeline gets cascaded into the next step, and thus the overall error is a reflection of the cumulative error across every stage in the pipeline. The final end to end accuracy, therefore, is expected to be lower than the accuracy at any of the individual stages. We have calculated the ratio of successful text-region zone pairs with ideal text-region zone pairs present in the inspection sheet. We define a successful text-region zone pair as the number of detected text-region with $IoU > 0.9$ mapped to arrows and subsequently to the correct zone. There are a total of 149 successful cases out of 179 cases, and hence the end to end accuracy is approx. **83.2%**. We also evaluate the accuracy of the final mapping, given annotated ROIs and arrow head and tail points on test set. In this case, there are total 165 successful cases out of 179 total cases, i.e. **92.1%** accurate as shown in last row of Table I.

TABLE I
RESULTS AT EACH STAGE AND INDIVIDUAL COMPONENTS.

	Results at each stage		Results of individual component	
Method	**Successful Cases**	**Accuracy**	**Successful Cases**	**Accuracy**
Arrow Detection (Faster-RCNN)	171 / 179	95.5%	171 / 179	95.5%
Text Mapping (Connected Components)	157 / 171	91.8%	166 / 179	92.7%
Zone Mapping (Template Matching)	162 / 171	94.7%	178 / 179	99.4%
Text to Zone Mapping (Overall Pipeline)	149 / 179	83.2%	165 / 179	92.1%

V. CONCLUSIONS AND FUTURE WORK

We have proposed a pipeline for the task of information extraction from manually tagged machine inspection sheets. Particularly, in the first stage of our pipeline, we suggest a novel combination of a template matching method, a deep learning based regression model and connected component analysis, to localize each specific components of the machine in the inspection sheet. In the second stage, we associate machine zones to detected text segments from the previous stage. The proposed method yields an accuracy of 83.2% at the end of the pipeline. In future, we plan to improve the proposed pipeline by utilizing more sophisticated and recent techniques such as detection via a deep attention model. In addition, it may be worthwhile to inject domain knowledge like the kind of faults that each zone tends to experience and eliminating cases for which a certain code or comment is not valid for the corresponding zone, into the proposed pipeline.

REFERENCES

[1] P. Ramakrishna, E. Hassan, R. Hebbalaguppe, M. Sharma, G. Gupta, L. Vig, G. Sharma, and G. Shroff, "An ar inspection framework: Feasibility study with multiple ar devices," in *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, Sept 2016, pp. 221–226.

[2] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497

[3] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 3476–3483.

[4] H. I. Koo and D. H. Kim, "Scene text detection via connected component clustering and nontext filtering," *Trans. Img. Proc.*, vol. 22, no. 6, pp. 2296–2305, Jun. 2013. [Online]. Available: http://dx.doi.org/10.1109/TIP.2013.2249082

[5] X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2. IEEE, 2004, pp. II–366.

[6] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Proceedings of the 2011 International Conference on Computer Vision*, ser. ICCV '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 1457–1464. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2011.6126402

[7] S. T. Anan Banharnsakun, "Object detection based on template matching through use of best-so-far abc," *Computational Intelligence and Neuroscience*, 2014.

[8] W. L. Nguyen Duc and Ogunbona, *An Improved Template Matching Method for Object Detection*. Springer Berlin Heidelberg, 2010, pp. 193–202.

[9] R. M. Dufour, E. L. Miller, and N. P. Galatsanos, "Template matching based object recognition with unknown geometric parameters," *IEEE Transactions on Image Processing*, vol. 11, no. 12, pp. 1385–1396, 2002.

[10] L. Wendling and S. Tabbone, "Recognition of arrows in line drawings based on the aggregation of geometric criteria using the choquet integral," in *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, Aug 2003, pp. 299–303 vol.1.

[11] S. Suchitra, R. K. Satzoda, and T. Srikanthan, "Detection & classification of arrow markings on roads using signed edge signatures," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*. IEEE, 2012, pp. 796–801.

[12] N. Wang, W. Liu, C. Zhang, H. Yuan, and J. Liu, "The detection and recognition of arrow markings recognition based on monocular vision," in *Control and Decision Conference, 2009. CCDC'09. Chinese*. IEEE, 2009, pp. 4380–4386.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Computer Vision and Pattern Recognition*, 2014.

[14] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *arXiv preprint arXiv:1507.05717*, 2015.

[15] C. Shi, C. Wang, B. Xiao, Y. Zhang, and S. Gao, "Scene text detection using graph model built upon maximally stable extremal regions," *Pattern Recogn. Lett.*, vol. 34, no. 2, pp. 107–116, Jan. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.patrec.2012.09.019

[16] A. Verma, M. Sharma, R. Hebbalaguppe, E. Hassan, and L. Vig, "Automatic container code recognition via spatial transformer networks and connected component region proposals," in *Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on*. IEEE, 2016, pp. 728–733.

[17] O. Vechtomova, "Introduction to information retrieval christopher d. manning, prabhakar raghavan, and hinrich schütze (stanford university, yahoo! research, and university of stuttgart) cambridge: Cambridge university press, 2008, xxi+ 482 pp; hardbound, isbn 978-0-521-86571-5," 2009.

[18] M. Shimrat, "Algorithm 112: Position of point relative to polygon," *Commun. ACM*, vol. 5, no. 8, pp. 434–, Aug. 1962. [Online]. Available: http://doi.acm.org/10.1145/368637.368653