


# Classification of Towels in a Robotic Workcell Using Deep Neural Networks

Jens Møller Rossen<sup>1</sup>, Patrick Søggaard Terp<sup>1</sup>, Norbert Krüger<sup>1</sup><sup>a</sup>, Laus Skovgård Bigum<sup>2</sup>  
and Tudor Morar<sup>2</sup>

<sup>1</sup>Maersk Mc-Kinney Moller Institute (MMMI), University of Southern Denmark (SDU),  
Campusvej, Odense, Denmark

<sup>2</sup>Inwatec, Odense, Denmark

**Keywords:** Image Classification, AI, Deep Neural Networks, Towels, Laundry Industry.

**Abstract:** The industrial laundry industry is becoming increasingly more automated. Inwatec, a company specializing in this field, is developing a new robot (BLIZZ) to automate the process of grasping individual clean towels from a pile, and hand them over to an external folding machine. However, to ensure that towels are folded consistently, information about the type and faces of the towels is required. This paper presents a proof of concept for a towel type and towel face classification system integrated in BLIZZ. These two classification problems are solved by means of a Deep Neural Network (DNN).

The performance of the proposed DNN on each of the two classification problems is presented, along with the performance of it solving both classification problems at the same time. It is concluded that the proposed network achieves classification accuracies of 94.48%, 97.71% and 98.52% on the face classification problem for three different towel types with non-identical faces. On the type classification problem, it achieves an accuracy of 99.10% on the full dataset. Additionally, it is concluded that the system achieves an accuracy of 96.96% when simultaneously classifying the type and face of a towel on the full dataset.

## 1 INTRODUCTION

Industrial laundries use many different processes when handling garments, e.g. washing, sorting and folding. Some of these processes have been automated, however others are still mainly manually processed. One such process is the feeding of towels to folding machines. While towel folding machines have been around for some time, automating the feeding of these machines still poses problems.

The problem itself is complex, since towels are non-rigid objects which requires complicated automatic manipulation. Furthermore, obtaining information about the type of towel being fed to the folding machine is critical, since the folding processes for different towel types may differ from one another depending on towel size, material and appearance (colors, patterns, logos). In particular, obtaining information about the appearance of towel types is critical to ensure consistent folding. This is because differences

in the appearance of the two faces of a towel causes different configurations after folding, as is illustrated in figure 1.

There already exists some robots that can automate the feeding of towels to folding machines (Sewts, 2023), however these robots lack the ability

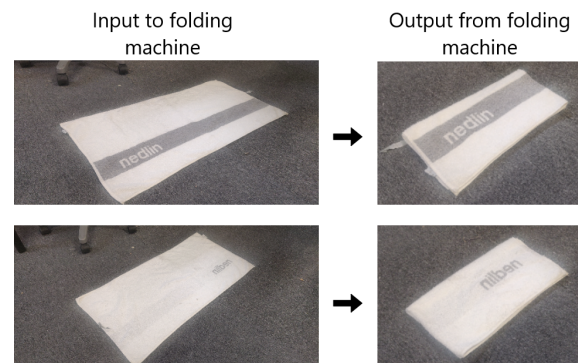


Figure 1: Illustration of how the result of the folding process of one type of towel is affected by which face of the towel is facing up on the conveyor when fed to the folding machine.

<sup>a</sup> <https://orcid.org/0000-0002-3931-116X>

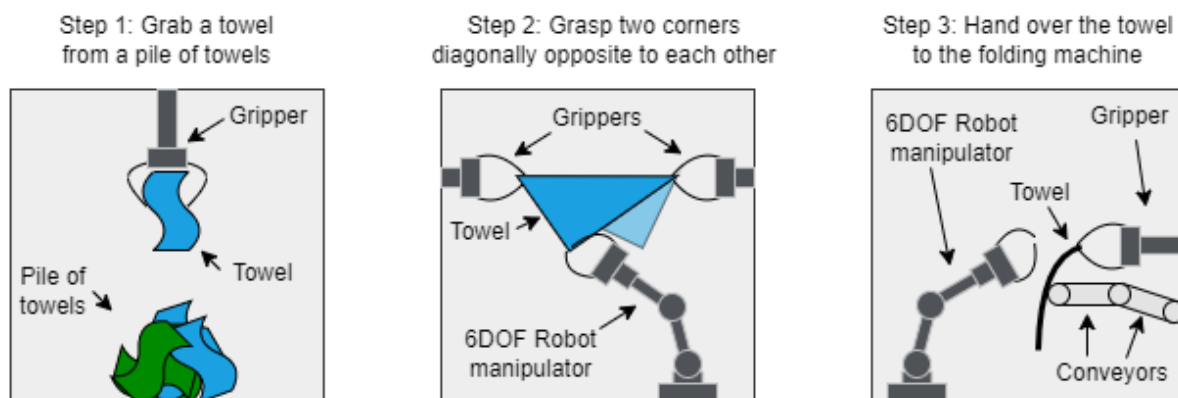


Figure 2: Illustration of how BLIZZ processes towels such that it can feed them to a folding machine. The illustrated process is divided into three steps. A descriptive text is found above each of the three steps.

to estimate the type and face of the towels they process which limits their functionality.

Likewise, the company Inwatec (Inwatec, 2023a) is developing a new robot named BLIZZ to solve this problem. It separates a towel from a pile of towels containing multiple different types, and subsequently feeds it to a folding machine. A simplified version of how it achieves this is illustrated in figure 2. However, like the aforementioned robots, it too cannot effectively determine what type of towel it is currently processing. This is referred to as the type classification problem. Furthermore, it cannot determine which face is the front and which is the back of a towel type, in the case that the two faces are different. This is referred to as the face classification problem.

In this work, we extend the functionality of BLIZZ through a computer vision based approach, such that it can solve both the type and face classification problems. The approach is split into two parts:

- 1) Designing a pipeline for collecting, processing and labeling images of towels.
- 2) Developing a DNN architecture that can classify the type of towel BLIZZ is currently processing, and — in case that the faces of the towel are different — classify which face of the towel is the front and which is the back.

## 2 STATE OF THE ART

Working with image classification typically involves both processing of images and designing, training and testing an AI network. Therefore, approaches in these two fields will be presented focusing on the problem of segmentation and classification of towels.

### 2.1 Image Processing

In the context of this paper, the processing of images concerns segmenting the towel from the rest of the image, i.e. the background. This process is generally known as image segmentation and is split into two main categories (Yu et al., 2023): Those that utilize classical segmentation methods and those that utilize Deep Learning (DL).

In (Maitin-Shepard et al., 2010), a classical segmentation method is used in a robot developed to fold towels. Images from two cameras are used to find corresponding depth-discontinuities along a towels edges after picking it up, thus allowing computation of grasping points. A foreground-background segmentation method is used to segment the towel, utilizing generated high precision background images.

In (Paulauskaite-Taraseviciene et al., 2022), a solution is presented to extract dimensions of garments like the size. They utilize the U-Net Convolutional Neural Network (CNN) architecture (Ronneberger et al., 2015) as a backbone to generate the masks used for segmentation.

Inwatec currently utilizes a classical foreground-background segmentation in BLIZZ similar to (Maitin-Shepard et al., 2010) to segment the towel from the background in multiple stages of towel processing. In this paper, a similar algorithm is used, but additional steps has been added.

### 2.2 Image Classification

For the problem of image classification, different learning methods have been applied. Since 2010 DNNs have become particularly popular for use in image classification problems, with the CNN architecture being especially efficient.

A CNN is used in (Gabas et al., 2016) to determine which of five possible classes a garment belongs to, which uses multiple depth images of the garment taken from different angles. The algorithm was tested on a dataset consisting of 4272 depth images. They used this dataset to train their own CNN network architecture. An average classification rate of 83% was achieved when classifying garments using only a single depth image, which increased to 92% when considering all images.

Inwatec currently utilize DNNs in their robots to solve image classification problems. In a master thesis written in collaboration with Inwatec (Lyngbye and Jakobsgaard, 2021), different CNN backbones were tested in combination with a custom fully connected top layer to classify different types of garments being separated by Inwatecs THOR robot garment separator (Inwatec, 2023b). The thesis achieved a classification accuracy of 97% on a dataset consisting of 17 different classes using ResNet (He et al., 2015) as a CNN backbone.

### 3 METHODS

This section presents the methods that have been developed to classify the type and face of towels processed by BLIZZ. The methods are split into two sections. The first section presents the designed image collection and processing pipeline, while the second section presents the developed DNN architecture used for image classification.

#### 3.1 Image Collection and Processing

To extract as much information as possible from a towel, it is desired to obtain two images, one of each face of the towel. To facilitate this, two stereo cameras (Intel D435) were installed in BLIZZ. Images of faces of a towel must be taken before BLIZZ has finished processing the towel, such that it can utilize the classifications output by the image classifier to manipulate the towel correctly. This, along with hardware constraints, meant that it was not possible to obtain complete images of both faces of towels. Instead, it was only possible to obtain two images, each one covering most of one face of the towel. An example of images captured by the two cameras can be seen in figure 3.

In order for the image classifier to achieve the highest possible classification accuracy, it is desired to remove the parts in both images in figure 3 that are not a part of the towel. This is done in a three stage process, which is shown in figure 4, where it is applied to the image shown in figure 3a.



(a) From 'Corner' camera.



(b) From 'Rear' camera.

Figure 3: Example of two captured images of a towel. Resolution of images is 1920 by 1080 pixels.

The first stage involves cropping the image according to a Region Of Interest (ROI), which is based on physical constraints of BLIZZ and size constraints of towels. The results of this stage are shown in the image in figure 4a.

The second stage involves segmenting the towel from the background in the image shown in figure 4a. This process is shown in figures 5a, 5b and 5c, with the result shown in figure 4b. The segmentation is performed by utilizing the depth image from the camera, where each pixel in the depth image contains the



(a) Stage I.



(b) Stage II.



(c) Stage III.

Figure 4: The results of each of the three stages in the image processing, applied to the image in figure 3a.

z-component of the Euclidian distance from the camera sensor to the object located at the pixel. An absolute difference is computed between the depth value at a given pixel and the depth value at the corresponding pixel in the background image, i.e., a depth image without a towel present. This absolute difference is computed for all pixels in the depth image, where each of the differences greater than a specified threshold is determined to belong to the towel, which yields a binary mask that is shown in figure 5a. To eliminate noise in the binary mask, only the largest contour is retained, which is shown in figure 5b. The edge of this contour is visualized as a red line in figure 5c.



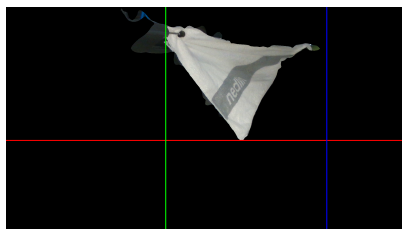
(a) Initial segmentation mask.



(b) After size filtration.



(c) After applying Sobel gradients.



(d) Calculated cropping lines.

Figure 5: Figures providing a detailed view of the process of towel segmentation (stage II) and cropping of the segmented image (stage III) as shown in figure 4.

Improving this segmentation is achieved by using Sobel gradients (Dawson-Howe, 2014). For each point in the contour, a line is fit to set of points. This set includes the point itself, along with the five preceding and proceeding contour points. Sobel gradients are then calculated along the normal of this line for a predefined length. The point is then moved to the position of the pixel that contains the highest Sobel gradient, since this is theorized to be the actual edge of the towel. The final contour can be seen in figure 5c, represented by the green line.

The third stage involves cropping the segmented image shown in figure 4b to the smallest bounding box that encompasses the towel. The lines that are used for cropping are shown in figure 5d, with the result of the cropping being shown in figure 4c. The cropping is performed using three cropping lines, which are calculated by two different methods. The first method computes two of the cropping lines using the image shown in figure 4b. The first of these is a vertical line, which goes through the non-zero pixel(s) with the highest x-value, while the second line is horizontal, and goes through the non-zero pixel(s) with the highest y-value. These cropping lines are visualized as the blue and red lines in figure 5d, respectively. The second method, which removes the gripper, also uses the image shown in figure 4b and is based on the position of the linear actuator grasping the towel. This method utilizes an estimated encoder to image mapping to calculate the cropping line visualised as a green line in figure 5d. All three of these cropping lines are then simultaneously applied to the image shown in figure 4b, which yields the image shown in figure 4c.

After images from both cameras, as shown in figure 3, have been segmented and cropped in the three stages process illustrated in figure 4, they must be further processed before they are usable to the image classifier. This involves scaling the images to  $500 \times 500$  pixels before concatenating them along the horizontal axis. This yields an image as shown in figure 6.



Figure 6: Example of an image of a towel after processing is finished. The towel in the image is of the type 'Nedlin', and is labelled as 'NedlinFront'.

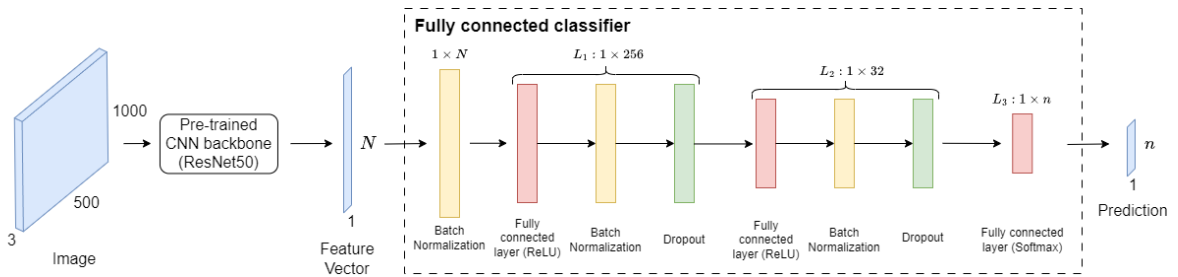


Figure 7: Architecture of the proposed network.

### 3.2 Image Classifier

The proposed image classifier utilizes the concept of CNNs based on the ResNet50 architecture (He et al., 2015) trained on the ImageNet dataset (Stanford Vision Lab, 2023) where the classifying layer is substituted with a custom Fully Connected (FC) classifier. This approach is common when designing CNN networks, and is generally known as transfer learning (Goodfellow et al., 2016). The particular architecture was chosen, since it proved the most effective in related work (Lyngbye and Jakobsgaard, 2021).

The reason for using an existing CNN is that they are trained on large datasets which in theory makes them good general feature extractors. Choosing the ResNet50 architecture was based on previous work in (Lyngbye and Jakobsgaard, 2021) as presented in the state of the art section, where different CNN backbones were tested on a similar image classification problem.

The architecture of the proposed image classifier is illustrated in figure 7. It takes a three channel  $500 \times 1000$  pixel color image of a towel as an input, as shown in figure 6. This image must be in the BGR-format and be normalized to the ImageNet dataset, since the ResNet50 architecture is trained on this dataset. This is done by converting images to the BGR-format and zero centering each of the color channels of the image with respect to the ImageNet dataset (Tensorflow, 2023).

The image is then processed by the ResNet50 architecture, whose output features are flattened to a feature vector of length  $1 \times N$ , where  $N$  is dependent

on the input dimensions of images. This feature vector is then passed to the custom FC classifier, which consists of two layers,  $L_1, L_2$  that contain 256 and 32 neurons, respectively. Additionally, each layer applies Batch Normalization (Ioffe and Szegedy, 2015) and Dropout (Srivastava et al., 2014) to the outputs of the fully connected layer respectively. This custom fully connected classifier yields a  $1 \times n$  vector containing the probabilities of the image belonging to each of the  $n$  classes in the dataset.

Apart from the data itself, the performance of the proposed image classifier is influenced by the value of the hyperparameters of the network. The hyperparameters seen in table 1 yield optimal performance of the image classifier. They have been determined experimentally by keeping all hyperparameter values constant, except the one being tested, and then training networks for 20 epochs on a subset of the full dataset using ten fold cross-validation.

## 4 DATASET

A dataset totalling 24120 images of six different types of towels, of which three have non-identical faces, has been collected. Illustrations of the different towel types are given in figure 8. An overview is given in table 2 where each row contains information about each of the towel types. In each row, it can be seen how many of the images have been labelled as 'Front' or 'Back' in their respective columns, in the case that the towel type has non-identical faces. If not, the 'I' col-

Table 1: CNN image classifier hyperparameters &amp; optimal values.

Hyperparameter	Optimal value
Image dimension	$500 \times 1000$
Learning rate	$1 \cdot 10^{-4}$
Batch size	32
Dropout rate	0.5
FC classifier layers	2
Layer neurons	256, 32

Table 2: Overview of the collected dataset and its contents.

Types	Front	Back	I	Total
'Nedlin'	1623	1750	-	3373
'BathTowel'	3342	3326	-	6568
'Rentex'	1673	1768	-	3441
'VDK'	-	-	6520	6520
'GrayStriped'	-	-	1176	1176
'YellowStriped'	-	-	1074	1074
Sum	6638	6844	8770	22152

umn shows how many images of that type is in the dataset. The 'Total' column shows how many images of a given towel type is in the dataset. The reason why the number of towels classified as 'Front' and 'Back' is not equal, is due to the stochastic nature of how towels are grasped by BLIZZ, where the probability for BLIZZ to pick up a towel which must be labeled 'Front' is equal to the probability to pick up a towel which must be labeled 'Back'.

Images are always given two labels. The first label, referred to as the type label, is related to what type of towel is present in the image. The second label, referred to as the face label, relates to the face



Figure 8: Illustrations of the six different towel types of which the collected dataset in table 2 is comprised.

visible in the left-half side of the part of the image captured by the 'Corner' camera (See figure 3a). In case of the image shown in figure 6, it is given the type label 'Nedlin', since the observed type matches the images displayed in figures 8a and 8b. Additionally, it is given the face label 'Front', since the face observed in the left-half side of the part of the image captured by the 'Corner' camera fits the image displayed in figure 8a.

## 5 RESULTS

Classification of the type of a towel is a multi-class classification problem, since there exists more than two different types of towels in the dataset presented in section 4. On the other hand, the face classification problem is inherently a binary classification problem, since towels of the same type whose faces are non-identical in the dataset have exactly two faces.

The proposed network has been trained and tested on both classification problems separately. The full dataset is used to test the type classification problem, thus containing  $n = 6$  different classes. For the face classification problem, the proposed network was trained and tested on three different datasets. Each of these datasets contains all images of only a single towel type whose faces are non-identical. This means that the proposed network was tested on three different datasets, each containing all images of the towel types 'Nedlin', 'BathTowel' and 'Rentex' respectively, and each dataset having  $n = 2$  classes.

All networks presented in this section have been trained, validated and tested, using an 80/10/10 split for 100 epochs, and the hyperparameters presented in

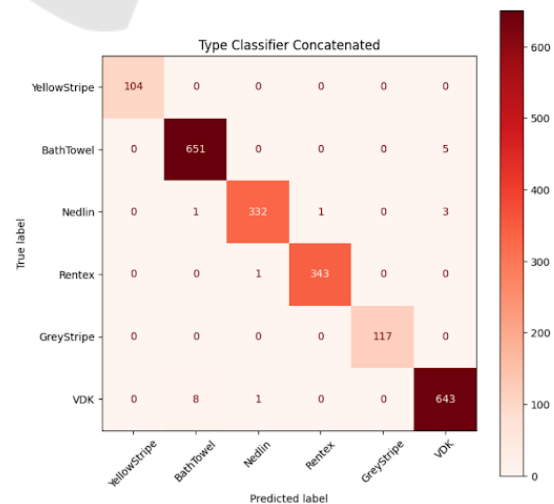


Figure 9: Confusion matrix for the network trained to classify types of towels. Network accuracy: 99.10%.

Table 3: Precision and recall values for the network trained to classify types of towels.

Towel Type	Precision [%]	Recall [%]
YellowStripe	100	100
BathTowel	98.64	99.24
Nedlin	99.40	98.52
Rentex	99.71	99.71
GreyStripe	100	100
VDK	98.77	98.62

Table 4: Accuracies of the networks trained to classify towel faces.

Towel Type	Accuracy [%]
BathTowel	97.71
Nedlin	98.52
Rentex	94.48

section 3.2. The categorical cross entropy function has been used to calculate the loss, with the exception of the network trained to solve the face classification problem, where the binary cross entropy loss function has been used instead. Furthermore, the Adam optimizer is used during training (Kingma and Ba, 2017).

The confusion matrix of the network trained to solve the type classification problem is shown in figure 9, from which the network accuracy is calculated to be 99.10%, while the precision and recall of the network on the individual types of towels is shown in table 3. The accuracy, precision and recall metrics have been calculated using the information found in (Evidently AI, 2023). Similarly, the accuracies for the network trained to solve the face classification problem on the three different datasets is shown in table 4, while the precision and recall for each towel type and face combination is shown in table 5.

A single network has also been trained on the full dataset presented in table 2 to solve both classification problems, meaning that it can determine both the type and face given an image of a towel. This is achieved by combining the type and face attributes of towels into a total of  $n = 9$  classes for the classifier to choose from. For example, a towel whose faces are non-identical would create two new classes, e.g. 'Nedlin-Front' and 'Nedlin-Back'. Meanwhile, a towel whose faces are identical would create only one class, e.g. 'VDK'. The performance of this network is shown in

Table 5: Precision and recall values for the networks trained to classify towel faces.

Towel Face	Precision [%]	Recall [%]
BathTowel - Front	97.00	98.48
BathTowel - Back	98.45	96.95
Nedlin - Front	98.01	98.67
Nedlin - Back	98.92	98.40
Rentex - Front	94.41	94.94
Rentex - Back	94.55	93.98

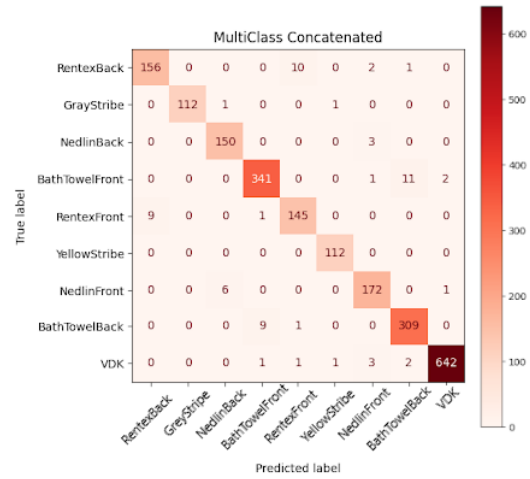


Figure 10: Confusion matrix for the network trained to classify both types and faces of towels. Network accuracy: 96.96%.

the confusion matrix in figure 10, while the precision and recall of each of the classes are shown in table 6.

## 6 DISCUSSION

When examining tables 4 and 5, it can be seen the the network trained to solve the face classification problem on a dataset consisting only of towels of the 'Rentex' type performs significantly worse than the two other tested networks. This is believed to be caused by a difference in the appearance of the towel, since the central logo on the 'Rentex' towel type (figures 8e and 8f) is more easily obscured on images than the off-center logo on the 'Nedlin' towel type (figures 8a and 8b), thus increasing the difficulty for the network to classify correctly. Additionally, this assumption is supported when comparing the 'Rentex' and 'BathTowel' towel types. Although the 'BathTowel' towel type dataset contains substantially more images than the 'Rentex' towel type dataset, its characteristic feature runs along the entire edge of the towel (fig-

Table 6: Precision values for the network trained to classify both towel types and faces.

Dataset class	Precision [%]	Recall [%]
RentexBack	94.55	92.31
RentexFront	92.36	93.55
NedlinBack	95.54	98.04
NedlinFront	95.03	96.09
BathTowelBack	95.67	96.87
BathTowelFront	97.15	96.06
GreyStripe	100	98.25
YellowStripe	98.25	100
VDK	99.53	98.77

ures 8c and 8d), thus almost guaranteeing that it will be visible when images are taken of that towel type.

As can be seen in the overview of the dataset in table 2 as well as in the confusion matrices in section 5 (figures 9 and 10), the dataset is imbalanced, with some classes containing more images than others. This is caused by the availability of towels during data collection, i.e. there were more towels available for some towel types than others, as well as the data collection process being automatic. It has been attempted to minimize the effects of an imbalanced dataset by matching the distribution of the full dataset when splitting it into training, validation and testsets. As seen in results of the network trained on the full dataset to solve the type classification problem (table 3), the imbalance doesn't seem to be a problem, since classes with a relatively large amount of data ('VDK') achieve comparable metrics to classes with relatively small amounts of data ('GreyStriped', 'YellowStriped').

## 7 CONCLUSION

In conclusion, this paper presents a proof of concept which is capable of capturing and processing images of towels being processed by Inwatecs BLIZZ using two mounted depth cameras. Furthermore, a CNN network has been developed, which classifies both the type and face of towels. This proof of concept can be used by BLIZZ to improve its functionality, enabling it to deliver towels to folding machines more consistently, while also improving its versatility.

A dataset consisting of six different types of towel, of which three have non-identical faces and totaling 22152 images has been collected and labelled. The developed image classification network has been trained and tested on this dataset, resulting in an accuracy of 99.10% when it is trained to solve only the type classification problem. Likewise, the proposed network trained to solve only the face classification problem achieves an accuracy of 94.48%, 97.71% and 98.52% on three different datasets consisting of images of just the 'Rentex', 'BathTowel' and 'Nedlin' towel types, respectively. Comparatively, when the proposed network is trained to solve both classification problems, it achieves an accuracy of 96.96%.

## REFERENCES

- Dawson-Howe, K. (2014). *A Practical Introduction to Computer Vision with OpenCV*. Wiley Publishing, 1st edition.
- Evidently AI (2023). Accuracy, precision, and recall in multi-class classification. <https://www.evidentlyai.com/classification-metrics/multi-class-metrics>. Accessed: 18-09-2023.
- Gabas, A., Corona, E., Alenyà, G., and Torras, C. (2016). Robot-aided cloth classification using depth information and cnns. In *Articulated Motion and Deformable Objects*.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Representation Learning*, page 536. MIT Press. <http://www.deeplearningbook.org>.
- He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.
- Inwatec (2023a). Inwatec. <https://inwatec.dk/>. Accessed: 14-09-2023.
- Inwatec (2023b). Thor. <https://inwatec.dk/products/thor-robot-separator/>. Accessed: 14-09-2023.
- Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift.
- Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization.
- Lyngbye, M. A. and Jakobsgaard, M. S. (2021). Garment classification using neural networks. Master's thesis, University of Southern Denmark, Odense, Denmark.
- Maitin-Shepard, J., Cusumano-Towner, M., Lei, J., and Abbeel, P. (2010). Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *2010 IEEE International Conference on Robotics and Automation*, pages 2308–2315.
- Paulauskaite-Taraseviciene, A., Noreika, E., Purtokas, R., Lagzdinyte-Budnike, I., Daniulaitis, V., and Salickaite-Zukauskienė, R. (2022). An intelligent solution for automatic garment measurement using image recognition technologies. *Applied Sciences*, 12(9).
- Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham. Springer International Publishing.
- Sewts (2023). Velum. <https://www.sewts.com/velum/>. Accessed: 14-09-2023.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.
- Stanford Vision Lab (2023). Imagenet. <https://www.image-net.org/>. Accessed: 14-09-2023.
- Tensorflow (2023). Resnet50-preprocess.input. [https://www.tensorflow.org/api\\_docs/python/tf/keras/applications/resnet50/preprocess.input](https://www.tensorflow.org/api_docs/python/tf/keras/applications/resnet50/preprocess.input). Accessed: 27-08-2023.
- Yu, Y., Wang, C., Fu, Q., Kou, R., Huang, F., Yang, B., Yang, T., and Gao, M. (2023). Techniques and challenges of image segmentation: A review. *Electronics*, 12(5).