Object Recognition in Python and MNIST Dataset Modification and Recognition with Five Machine Learning Classifiers

Yordanka Karayaneva and Diana Hintea
Faculty of Engineering, Environment and Computing, Coventry University, Coventry, United Kingdom
Email: karayany@uni.coventry.ac.uk, ab8351@coventry.ac.uk

***Abstract*—Schools in many parts of the world use robots as social peers in order to interact with children and young students for a rich experience. Such use has shown significant enhancement of children's learning. This project uses the humanoid robot NAO which provides object recognition of colours, shapes, typed words, and handwritten digits and operators. The recognition of typed words provides performance of the corresponding movements in the sign language. Five classifiers including neural networks are used for the handwritten recognition of digits and operators. The accuracy of the object recognition algorithms are within the range of 82%-92% when tested on images captured by the robot including the movements which represent words in the sign language. The five classifiers for handwritten recognition produce highly accurate results which are within the range of 87%-98%. This project will serve as a promising provision for an affective touch for children and young students.**

***Index Terms*—object recognition, image processing, machine learning, robotics, visual learning**

I. Introduction

Robots aid children and young students at educational institutions throughout the world and such integration has shown substantial results [1]. The aim of this project is to develop an object recognition application tested on the humanoid robot NAO which can provide means for future development and actual use in institutions. The algorithms are written in Python and they include image pre-processing techniques with the consequent use of functions from the libraries Python Image Library (PIL), OpenCV (Open Source Computer Vision Library), and tesseract. The robot is able to recognise colours, shapes, typed text, and handwritten digits and operators the latter of which form simple mathematical problems. In addition to this, the robot also performs movements in the sign language corresponding to a written word that is successfully recognised.

Once the shapes are recognised, NAO points at the two figures one by one with saying their type and colour. As humanoid robots are used in schools mainly for teaching the young to write code or learn a foreign language, this study can be classified as novel due to the fact that it presents innovative means for children's visual learning enhancement.

Manuscript received February 16, 2018; revised June 22, 2018.

II. Related Work

A. Object Recognition Algorithms

Image processing has been used for decades in a wide range of areas including business, medicine, communication and robot automation, the latter being the topic of this study. In terms of object recognition, the aim is to detect a certain object or a property in an image as it belongs to a class of objects. Recognition can be performed on faces, colours, shapes, text and others. According to Amit and Felzenswalb [2], there exist two major types for object recognition: generative and discriminative. Generative object recognition involves building a probability model which determines what objects belong to the desired class and the ones that are irrelevant. The obtained model is related to estimating certain parameters of the pose which are based on probabilities. Discriminative object recognition takes advantage of a classifier that can distinguish between images with the desired objects and images that do not contain it. A common use of the latter type involves machine learning algorithms such as neural networks for training and testing. Absolutely accurate algorithms do not exist due to images constraints such as lightning, positioning and rotation stated by Khurana and Awasthi [3]. Lighting can have impact both on indoor and outdoor images bearing in mind the weather and lights. Positioning refers to the position of the object in an image which is not expected to be a considerable issue. Rotation involves whether the object is straight in the image or not which was an issue in the past, but novel algorithms can easily tackle this constraint and perform successful object recognition.

B. Object Recognition with OpenCV in Python

OpenCV (Open Source Computer Vision) is a powerful image processing library written in C/C++, and it is available for Python. It is widely used for object recognition based on colour and shape. Object recognition of a red colour has been described by Kurukshetra [4]. The first procedure that is applied is image segmentation-preprocessing algorithm in order to reduce undesired distortions and enhance image data. Thus, the object is

doi: 10.18178/joig.6.1.10-20

isolated from the background based on a critical value or a threshold.

C. Machine Learning Classifiers

In relation to machine learning classification, the algorithms that are taken into account are Random Forest, Stochastic Gradient Descent, Support Vector Machine, Nearest Neighbours and Neural Networks. Random Forest is an algorithm which builds multiple classification. In order to classify a new input vector, it is placed in each of the trees in the forest, and 'voting' takes place. The classification that is selected is the one that received the most 'votes'. Furthermore, Random Forest provides highly accurate results and efficiency on large datasets [5]. The algorithm Stochastic Gradient Descent is popular for large-scale learning problems. This algorithm does not compute the gradient but instead, at each iteration the gradient is estimated based on a single or a few examples [6]. Support Vector Machines are supervised machine learning models that are generally used for classification problems. They use a classification hyperplane which discriminates between different classes [7]. K-Nearest Neighbours algorithm separates data points in different classes in order to classify a new sample point. Once, a new observation needs to be predicted, the k-nearest data points are taken into account and the most appropriate is selected [8]. Artificial neural networks are models that are biologically inspired by the human's brain. They consist of an input, hidden and output layers and have processing units called 'neurons' or 'cells' which communicate by sending signals to each other. During supervised learning, the network is trained with the provision of both input and output patterns, where in the latter the categories for the classification are specified [9].

D. Classification Methods Used on MNIST Dataset

MNIST is a dataset which contains handwritten images of all digits 0-9. The training set was created with the use of the handwriting of employees of the American Census Bureau while the testing set contains American high school students' handwriting. MNIST is a very popular dataset which brings the attention of researchers and usually the error rate is used for estimation of the success of the machine learning methods that have been used. The error rate can be calculated easily once the accuracy has been obtained, *(errorrate = 1 − accuracy)* [10]. In terms of accuracy, it is easily estimated with the use of the testing set for each of the classifiers once they are generated and saved.

The most commonly used classification methods on this dataset by other researchers are Support Vector Machine (SVM) and Artificial Neural Networks. In a recent paper Khan [11] uses Support Vector Machine that achieves an accuracy of 99.36% for the test set which has the error rate of 0.64% (2017). This represents a better performance than the performance of the same classification method of the creators of the dataset listed in their original paper where they achieve an error rate of 0.8% [12].

Random Forest classification was applied to the MNIST recognition problem where the researchers used two ranges for the number K of random features and the number L of trees [13]. For K the two ranges were from 1 to 16, and from 20 to 84. In relation to L, six increasing values were picked, from 10 to 300 trees. The most successful recognition was obtained from two tests with K = 12 and L = 200, and K = 13, L = 300, with an accuracy rate of 93.24%.

In terms of K-Nearest Neighbours classifiers, Hamid and Sjarif [14] have achieved an accuracy of 99.26% for k = 1 to 15 where the error rate is logically 0.74%. When the number of neighbours grow larger than 15, the accuracy begins to drop on a small scale.

Optimisation algorithms can be applied to machine learning classifiers in order to increase their performance. Tuba E., Tuba M., and Simian [15] used a Bat optimization algorithm in order to enhance Support Vector Machine for the handwritten digits recognition. The researchers intentionally use weak features where other machine learning methods will not derive satisfactory results. The achieved accuracy on the testing set is 95.6% which according to the authors, is better than other methods which use a weak features set including a multilayer neural network, extreme learning machine, regularized extreme learning machine, and optimal weight learning machine.

The machine learning classifiers that produced the most significant results are Artificial Neural Networks (ANNs). Multi-column deep neural networks were used for the handwritten digits recognition, and the researchers claim to be the first that have achieved a "near-human" performance with an error rate of 0.23% [16]. The lowest know error rate that has been achieved on the MNIST dataset is 0.21%. Two research projects claim to have achieved this performance including project that takes advantage of regularization of neural networks with the use of DropConnect [17], and a project from the Parallel Computing Center in Ukraine which ensembles five convolutional neural networks for this purpose [18].

E. Implementation of Object Recognition Algorithms on NAO Robot

A few studies investigate how object recognition algorithms have been implemented on NAO robots. A study by Wenbai *et al.* [19] describes a human gesture recognition where the robot takes video of the gesture performed by a human in front of it, and then tries to recreate it. Object recognition algorithms with the use of deep learning approach have been implemented by Albani *et al.* [20]. The NAO robot was trained to recognise particular objects such as a ball as the purpose of this study was to allow the robot to participate in the annual football competition for robots named RoboCup. Pre-trained Convolutional Neural Networks were used so the robot could detect an object. A dataset was used which contained a significant number of real world pictures. The images were captured by different NAO robots (players) in a variety of fields. This study has been considered successful due to the high accuracy of the object recognition algorithms.

F. Introduction of NAO Robot in Schools

Robots have been used in educational institutions in order to provide the school community with an innovative and interesting method for teaching and learning. The 'subject' that the robot teaches, and the role it has need to be considered prior to developing any applications. The subject can be technical such as robotics and computer science or non-technical which includes teaching a foreign language or mathematics. The latter is more applicable to children and young students. In terms of their role, the robot can be passive as in the examples of technical subjects, or other roles include teaching tool, co-learner, companion and care receiver. NAO robots have been widely used in many parts of the world for children's learning enhancement. It was used in a Danish school for programming, learning a foreign language and poems development [21]. *Altin et al.* [22] describe how the NAO robot has been used in order to teach students to write letters from the alphabet. Object recognition algorithm has been used for handwriting which is a further step more advanced than typed text. Sign language movements based on object recognition algorithms with OpenCV have been used by Ertugrul *et al.* [23] in order to help autistic children better express themselves. Another experiment conducted recently shows an alternative approach in terms of the role of the robot and the children [24]. Children show cards with words written in English to the robot which does not translate the word, but explains the word with gestures. Thus, children are expected to discover the word by listening to the robot and its explanation.

III. Methodology

A. Prototyping Methodology

The methodology used in this paper is the Prototyping methodology which is appropriate for code development and integration. This methodology stresses on the implementation of early increments which are further refined in the later stages of the development process. In terms of requirements gathering, the objectives of the project and its scope are defined based on the literature review of already existing studies. The analysis stage considers how these objectives could be achieved and Python libraries are taken into account including OpenCV, PIL, tesseract etc. The implementation of the algorithms involves editing the captured image in order to perform noise removal and better quality. Next, algorithms are tested initially on images obtained from the web, and then they are captured by NAO robot in order to estimate whether they work or not. If the algorithms are unsuccessful, the process returns to the analysis stage for further evaluations on how it could be improved. The final stage validation imposes on the fact that the algorithms need minimal or zero adjustments and further tests for their overall accuracy on a pre-defined number of tests can be estimated in order to judge their quality. (Fig. 1)

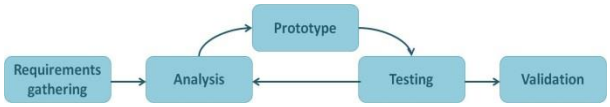

Figure 1. Prototyping methodology

B. Implementation of Object Recognition Algorithms

The algorithms that form the entire architecture are typed text recognition with sign language movements, colour recognition, shape recognition, and handwritten digits and operators recognition with the use of five machine learning classifiers. This innovative and interesting approach is targeted at children for educational purposes. The reasons behind the choice of these particular algorithms is within the fact that a small amount of time is required for their execution which is coherent with the impatience of younger students and especially children. Thus, they will not have to dedicate a huge amount of time waiting for the robot to interact with them in relation to the nature of the algorithms including speech and movements. Furthermore, the aforementioned algorithms have highly considerable accuracy which makes them relevant to be taken into account as in a real-world setting, the amount of errors needs to be minimized.

The intention of a potential introduction of the architecture in a classroom will include multiple robots with a client/server architecture which will allow a teacher to monitor the robots' interaction with the children. Thus, the teacher will need to have basic knowledge of how to run the algorithms and give instructions to the students when this happens. However, the teacher will not have to be an expert in this field rather than having primary concepts in mind. (Fig. 2)

Figure 2. Examples of digits and operators

The Python library tesseract is used for the text detection algorithm. The image needs to be cropped initially in order for the built-in function in the library to work. Firstly, the largest contour area is detected in the image. A mask is created in order to select only the part of the image that contains text. The area outside the mask is converted to white, and then, a new image is obtained. For the avoidance of small noise areas, any contours with a size smaller than a pre-defined value are removed. Then, a built-in library function is used for the text detection. The movements of NAO in the sign language that follow are implemented using joints which represent different parts of the robot's body with the corresponding coordinates. In terms of shape recognition, the desired image is converted to a black and white image, and then

to a binary image. Next, x-min, x-max, y-min and y-max contours are found and the image is cropped in order to obtain only the figures without the surroundings. Then, a consideration of the number of vertices of the contours is made. Sequentially, if the number of the vertices equals three, the shape is classified as a triangle. Similarly, if the number is four, the shape is either a square or a rectangle. In order to judge between the two, their sides have to be compared, and if the difference is more than 10 pixels, it is concluded that the figure is a rectangle. For the recognition of the colour of the shape, Lab colour space is taken advantage of. Then, a built-in OpenCV function is used in order to find the contours. Bearing in mind the fact that the image has been resized and made smaller, if the accurate coordinates of the centre are to be found, a multiplication by ratio the coordinates of the edited image has to be done. Ratio is the proportion of the original image to the edited one. Then, add x-min and x-max are added. As for colour recognition, Python Image Library (PIL) is used. The image is cropped in order to obtain a 100 x 100 image with the centre of the original image. Then, the average of the colour is estimated by using the RGB (Red, Green, Blue) colour scheme where which colour has pre-defined value for R, G and B. The average for these values is therefore found with the use of a function. Consecutively, the nearest colour is to be found with another built-in function which is later converted into a 6-digits number. The final step is to find the actual name of the colour with the use of a vocabulary function.

C. Implementation of the Movements in Sign Language

In terms of sign language expressions, the gestures were implemented with the use of joints which represent different parts of the robot's arm and hand. This involved the coordinates of the joints which define the level of stretching along with speed of execution. In relation to the gestures of sentences, certain words are not taken into account such as 'the', 'a', and 'an'. This is why they were omitted in the algorithms as they do not have any gestures representations.

D. Implementation of an Algorithm for the Robot to Point at Figures

In order for the robot to be able to point at the figures, the coordinates of the centre of each shape on the sheet are retrieved. The coordinates are then converted to the actual coordinates which are used for the robot to point at the centre of the figures with its arm bearing in mind that there are usually two shapes on a single sheet. The robot pronounces the type of one of the shapes along with its colour and it points at it. The same approach applies to the other shape.

E. Implementation of the Machine Learning Algorithms

The machine learning classifiers are trained and tested on the dataset MNIST which contains 60,000 examples for training and 10,000 examples for testing of handwritten digits from 0 to 9 [25]. Another 4,000 handwritten pluses (+) and 4,000 handwritten minuses (-) are included to the database. Each example is an image with the size 28 x 28. There are two scripts implemented, one for training, and another for testing. In relation to the script for training, histogram of oriented gradients (HOG) is calculated for each image. Then, the classifier is saved in a file. The script for testing involves images where basic mathematical expressions are written by hand. The image needs to be edited for the classifier to work by cropping the images so only the expression remains. For better results, the digits and the operators need to be written solid without any white areas in them. Otherwise, it may not derive very successful results.

F. Pre-processing Techniques on the Images

In order to perform recognition on actual images that contain handwritten expression, pre-processing of the images is conducted which contains several steps with the use of functions in the computer vision library named OpenCV.

A technique that this research project implements is the Histogram of Oriented Descents (HOG) which is an image descriptor successfully used for machine learning and computer vision. The scikit-image library in Python has already implemented the HOG descriptor with the use of a feature sub-package [26]. Furthermore, a few arguments need to be passed in including the number of orientations, pixels per cell, cells per block, and whether or not the square-root transformation should be applied to the image prior the process of computing the HOG descriptor.

When taking advantage of machine learning for a specific problem, some of the features used for the classification or regression may seem irrelevant. This is why, feature selection is performed in order to simplify the model and decrease the running time and memory usage [27]. In this research, the colour of the image is not relevant to the recognition of the digits and operators, and hence, the image is converted from RGB (Red, Green, Blue) to a grayscale image. This process eases the image processing and now each pixel is represented by exactly one value in the range of 0 to 255 where 0 represents black, 255 represents white, and the values between those two represent shades of grey.

Then, blurring is applied to the image which is an important method used in image processing for thresholding and edge detection due to the fact that it increases their performance. The blurring technique that is used in this paper is known as Gaussian blurring where instead of using a simple mean, a weighted mean is applied where neighbourhood pixels that are closer to the central pixel, contribute more to the "average". The final result of this technique of blurring is that the image is less blurred than using other techniques such as average blurring, but also more naturally blurred. (Fig. 3)

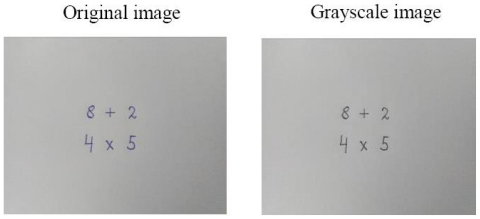

Figure 3. Conversion of an RGB image to a grayscale image

The next step is to crop the separate expressions. The algorithm is applicable for a multiple number of expressions written on a piece of paper, and as it is observed in the current example, there exist two of them.

Figure 4. Cropping of the two expressions

The subsequent steps are explained with the use of the first expression only. After the expression has been cropped from the initial image, the characters themselves need to be partitioned which is known as image segmentation [28]. This process in mandatory as the separate characters are to be recognised individually. (Fig. 5)

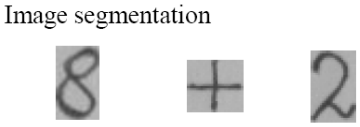

Figure 5. Partitioning of the first expression

The process that follows is called extraction of features and the aim of it is to transform the format of the images derived from segmentation to the format that is used in the MNIST dataset [29]. As it is already mentioned, the size of the images used in the dataset is 28 x 28 pixels. Furthermore, the digits itself has to be placed exactly at the centre of the image. The obtained images will rarely have this specific size, and thus, it has to be further adapted in order to be usable by the classification methods. Finally, the contours of the edged image are found, which are sorted from left to right. Each of these contours represent a digit in the image that has to be classified. Then, a bounding box is computed for the contours which the OpenCV built-in function cv2.boundingRect returns the starting coordinates (x,y) of the bounding box with the corresponding width and height of it. Additionally to this, the width and height are checked in order to ensure that the box is at least seven pixels wide and twenty pixels tall. If the bounding box does not meet these dimensions, it is considered too small to be an actual digit. If these dimensions are satisfied, the Region of Interest (ROI) is extracted which holds the digit to be classified [30].

A pre-processing step however has to be applied prior the process of an actual recognition. Thresholding has to be applied which is a binarization of the image. The purpose of it is to convert a grayscale image to a binary image. A binary image is composed of only two colours, black and white. Black is represented by the value of 0, while white is represented by the value of 255. The idea behind the thresholding method is to set all pixels below a specific value called thresholding value T to be 0, and if they are greater than this value, to 255. The thresholding technique that is used in this research is known as Otsu's method [31]. Unlike simple thresholding which is a generic method, the thresholding value is computed automatically. If the T value has to be defined manually, a wide range of experiments have to be conducted prior this selection. Another problem is that the image itself may exhibit a huge number of pixel intensities which decreases the level of suitability of the simple thresholding method. In other words, if there exist only one value of T, it may not be sufficient. Otsu thresholding overcomes this problem by assuming that there are two peaks in the grayscale histogram of the image. Then, it tries to find an optimal value to separate these two peaks, which appears to be the value of T.

After the use of Otsu's thresholding method, the digit is then deskewed and translated to the centre of the image. Then, the HOG feature of the thresholded ROI is computed. The HOG feature is fed to the classifier's predict method which determines which digit the ROI is. (Fig. 6)

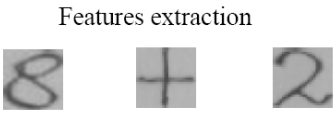

Figure 6. Conversion of the images to the needed format

Features extraction was the final step prior the actual testing of the derived images with the four classifiers created earlier. If the classification is successful, the result that is to be obtained is represented in a string ("8 + 2"). In order to calculate this expression, the string has to be converted to a mathematical expression. This is achieved by the use of two built-in functions in Python – a function named 'join' for the removal of spaces between the characters, and a function named 'eval' for the actual transformation of a string to an expression. Once the string has been successfully transformed to an expression, it is calculated and an answer is finally provided by being printed on the screen. In order to make the classification more professional, the answer can be displayed on the image itself. An OpenCV built-in function is used named cv2.regctangle which draws a green rectangle around the current digits and operators. The cv2.putText is used which draws the digits and operators with an equation sign, and the final answer. The first argument of this function is the image to draw, while the second argument represents the string already described. The next argument defines the location of the drawn text which are defined to be above the handwritten expression. The fourth arguments depicts the font of the text, and the final fifth argument defines the colour which in this case is chosen to be pink. (Fig. 7)

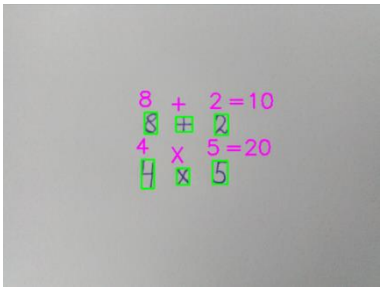

Figure 7. Successful recognition of all characters in the two expressions

G. Establishing Connection with the Physical Robot

Testing limited areas of the algorithms can be done in the simulators Choregraph or V-REP as they do not support all features. These simulators appear to be ideal for the testing of the movements in sign language only. In order to test the algorithms on the physical robot, a connection needs to be established using NAO's IP address and port. The image captured is stored in its own location. Thus, the library paramiko is used in order to retrieve this image from the remote served using the IP address, username and password [32]. Once the image is retrieved, it is stored in a local path in order to use it in the implemented algorithms.

IV. Evaluation

A. How the Experiments Were Performed

NAO robot has two cameras which allow it to capture pictures and record a video. For this experiment, capturing pictures is used with the upfront camera. The type of camera can be specified together with the format and the name of the image. Once the picture is captured, it is retrieved with paramiko library and it is successfully saved in the same folder with the Python code. It is usually saved under the name image.jpg which can be optionally altered, and then the Python code works with the same image. A sheet with the needed object is held in front of the robot's camera which is located on its forehead. By using the simulator Choregraph, the robot's vision can be observed in order to better place the sheet. Then, the Python script could be run which establishes connection with the robot using its IP address and port, captures a picture which is then retrieved, and works on the same image.

B. Typed Text Recognition and Sign Language Gestures

In relation to typed text recognition, the algorithm is adapted to detect the contours of the text, and the recognised word is stored in a string variable. Simple thresholding and Adaptive Gaussian Thresholding functions available in OpenCV are used as the latter shows successful recognition in cases when Simple Thresholding fails to produce a correct output. This is due to the fact that the brightness of the image is darker which causes issue to identify the contours of the image. (Fig. 8)

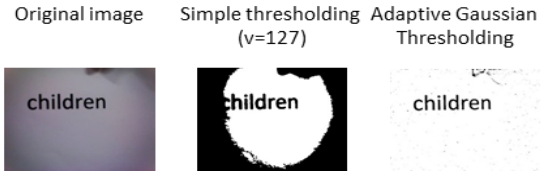

Figure 8. An image recognized by another method

The two thresholding functions use an optimal threshold value T. All pixels intensities below T are set to 255, and similarly, the ones that are greater than T are set to 0. Gaussian blurring is applied to the images prior to using the thresholding functions. Furthermore, Simple Thresholding is limited in terms of the fact that the T value has to be provided by the user, and a good value may take a considerable amount of time and experiments. In order to overcome this problem, Adaptive thresholding is used which finds an optimal threshold value T by considering small neighbours of pixels. This method is particularly useful in situations where there exist huge ranges of pixel intensities and the T value may change for the different parts of the images [33]. The algorithm has been tested on 100 images, and it showed a considerable accuracy of 91%. The movements in the sign language involve only the arms of the robot and hence, they are easily performed with zero risks involving losing balance. A small dataset of words has been implemented which can be grouped into short sentences. The robot is also able to perform the sentences by omitting the use of certain words that are already specified in the methodology chapter. As part of future work, the dataset can be further enriched. Due to the fact that the robot possesses different arm/hand joints such as wrist, elbow and shoulder, the movements are easily observed, as they resemble a near human-like representation of the sign language words. (Fig. 9)

Figure 9. Recognition of the word teacher and sign language gestures

C. Colour Recognition

Colour recognition has shown considerable results in the majority of cases with an accuracy of 89% however, it still depends on the lighting conditions in the room. The image is initially cropped and then PIL is applied in order to recognise the primary colours or any shades of them. Due to the lighting conditions, it is common for the algorithm to wrongly recognise primary colours as their shades. This can be partly overcome by replacing the strings in the code with the corresponding primary colour, but it will prevent a correct recognition of shades when they are presented. (Fig. 10)

Figure 10. Recognition of the colour red

D. Shape Recognition

As for the shape recognition, the algorithm is adjusted to work for two shapes on a piece of paper together with their colours. Likewise colour recognition, this algorithm also depends on the lighting conditions in the room as there are examples with slightly darker areas of the pictures where one of the figures is not recognised.

Once the shapes are recognised, the robot points at them one by one with its hand. Due to the fact that the shapes are small in size, this cannot be distinctly observed.

The accuracy of the shape recognition algorithm is 82% over 100 pictures while the accuracy for the gestures is of the same value as when the shapes are correctly recognised, they are followed by the arm and hand movements. (Fig. 11)

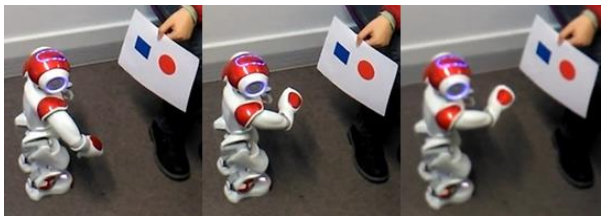

Figure 11. Shape recognition

E. Handwritten Digits and Operators Recognition with Machine Learning Classifiers

In order to evaluate the performance of the four classifiers, tests are performed including accuracy and error rate estimation, confusion matrices and visualization of predicted labels versus actual labels. The number of the used examples from the testing set are 1000. Further tests in relation to actual images recognition are conducted as well which are rarely covered in previous research.

Confusion matrices give important information in relation to the number of correctly and incorrectly recognised examples [34]. In terms of the incorrect recognitions, it can be observed when a particular digit has been recognised as another, including the number of cases. The confusion matrices in this research show the recognition of all 10 digits from 0 to 9, and additionally the 11th character is a plus ('+'), the 12th character is a minus ('-'), and the final 13th character is an '×' used for multiplication.

In relation to another approach of an estimation of the classification methods' performance, Actual versus Predicted label is used as scatter plots. This illustration is a rich form of data visualization by which the overall quality and performance of the created model can be judged and estimated [35].

F. SVM with Gaussian Kernel Evaluation

The confusion matrix for Support Vector Machine with a Gaussian kernel shows that only '0' and '×' are predicted in all cases while the other characters are misclassified in a small minority of cases. (Fig. 12 and Fig. 13)

```
Confusion matrix - SVM:
[[77  0  0  0  0  0  0  0  0  0  0  0]
 [ 0 84  0  0  1  0  0  0  0  1  0  0  0]
 [ 0  0 69  2  1  0  0  1  2  0  0  0  0]
 [ 0  0  4 73  0  1  0  1  2  0  0  1  0]
 [ 0  0  1  0 58  0  4  0  0  3  0  0  0]
 [ 1  0  0  2  0 71  1  0  2  1  0  0  0]
 [ 0  1  0  1  1  0 71  0  2  1  1  0  0]
 [ 0  2  1  0  0  0  0 67  0  1  3  0  0]
 [ 0  0  1  0  1  1  0  3 76  1  0  0  1]
 [ 0  1  0  3  1  0  1  3  1 74  0  0  0]
 [ 0  0  0  0  0  1  0  3  0  0 71  0  0]
 [ 0  0  1  0  0  2  0  0  0  0  0 79  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0 59]]
```

Figure 12. Confusion matrix for support vector machine

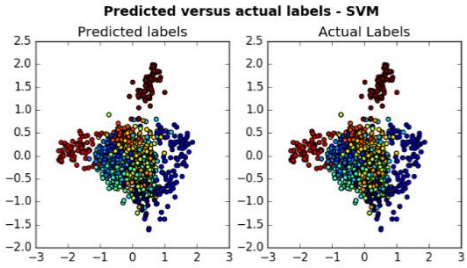

Figure 13. Scatter plot for SVM

G. Random Forest evaluation

When using a Random Forest classification method, the plus '+' the '×' characters are completely recognised while there exist a small number of confusions in relation to the remaining characters. (Fig. 14 and Fig. 15)

```
Confusion matrix - RF:
[[76  0  0  0  0  0  0  0  0  1  0  0  0]
 [ 0 84  0  0  1  0  0  0  1  0  0  0  0]
 [ 0  0 70  2  0  0  1  1  1  0  0  0  0]
 [ 1  0  3 73  0  1  0  2  1  0  0  1  0]
 [ 1  0  0  0 59  0  4  0  0  2  0  0  0]
 [ 2  0  1  2  0 68  1  0  4  0  0  0  0]
 [ 1  1  0  1  3  1 69  0  1  0  1  0  0]
 [ 0  1  4  0  0  0  0 63  0  3  3  0  0]
 [ 0  0  1  1  1  0  1  1 75  3  0  0  1]
 [ 0  1  0  1  0  1  1  3  1 76  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0 75  0  0]
 [ 0  0  1  0  0  1  0  1  0  0  0 79  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0 59]]
```

Figure 14. Confusion matrix for random forest

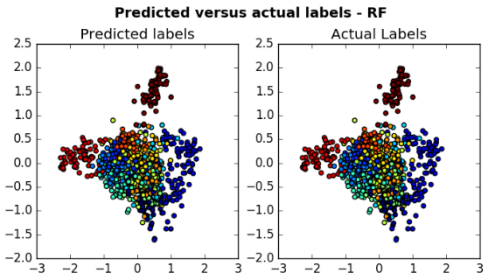

Figure 15. Scatter plot for random forest

H. K-Nearest Neighbours Evaluation

K-Nearest Neighbours' confusion matrix is similar to the matrix of Support Vector Machine with a Gaussian kernel. The '0' and '×' characters have 100% recognition while minor misclassifications are present with the remaining characters. (Fig. 16 and Fig. 17)

```
Confusion matrix - KNN:
[[77  0  0  0  0  0  0  0  0  0  0  0]
 [ 1 84  0  0  1  0  0  0  0  0  0  0]
 [ 2  0 66  3  0  0  0  1  1  0  1  0  1]
 [ 2  0  3 72  1  2  0  1  1  0  0  0  0]
 [ 1  0  0  0 58  0  3  0  0  4  0  0  0]
 [ 2  0  0  3  0 66  1  0  6  0  0  0  0]
 [ 1  1  0  1  1  0 70  0  2  1  1  0  0]
 [ 0  1  4  0  0  0  0 62  0  4  3  0  0]
 [ 1  0  1  1  1  0  1  0 76  2  0  0  1]
 [ 0  0  1  1  0  1  2  1  1 77  0  0  0]
 [ 0  0  0  0  0  1  0  1  1  1 71  0  0]
 [ 0  0  1  2  0  3  0  2  0  0  0 74  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0 59]]
```

Figure 16. Confusion matrix for K-nearest neighbours

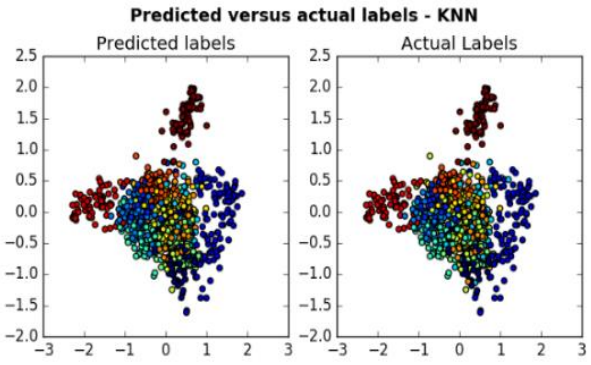

Figure 17. Scatter plot for K-nearest neighbours

I. *Linear SVM with SGD Training Evaluation*

Stochastic Gradient Descent optimization algorithm for training is used for a Linear Support Vector Machine which has the lowest accuracy which also has a significant negative difference with a comparison with the other three classification methods. The low performance of this classifier can be easily noticed on the confusion matrix and the scatter plot. An important observance from the confusion matrix is the considerably low recognition of the digit 9 which is often misrecognized as 3, 7 and 8, and in some cases it is interpreted as 0, 1, 4, 5, 6. However, there is full recognition of the character '×' likewise the other classifiers, and 0 is misclassified only once as 4. (Fig. 18 and Fig. 19)

```
Confusion matrix - SGD:
[[76  0  0  0  1  0  0  0  0  0  0  0  0]
 [ 0 84  1  0  0  0  1  0  0  0  0  0  0]
 [ 0  0 62  4  0  0  1  2  5  0  0  1  0]
 [ 2  0  3 67  0  1  0  1  6  1  0  1  0]
 [ 2  0  0  3 50  0  3  0  6  2  0  0  0]
 [ 1  0  0  2  0 68  3  0  4  0  0  0  0]
 [ 0  1  0  0  1  2 70  0  3  1  0  0  0]
 [ 1  1  3  2  1  0  0 56  3  0  6  1  0]
 [ 0  2  1  2  2  5  4  2 58  5  2  0  1]
 [ 4  3  0 10  1  5  1  6  6 48  0  0  0]
 [ 0  0  2  2  0  0  0  4  4  0 63  0  0]
 [ 0  0  2  3  0  1  0  4  0  0  1 71  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0 59]]
```

Figure 18. Confusion matrix for linear SVM with SGD training

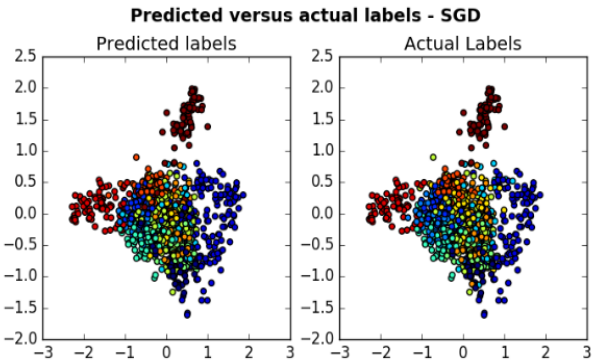

Figure 19. Scatter plot for linear SVM with SGB training

J. *Estimation of Methods' Accuracy Using Two Types of Testing (Fig. 20)*

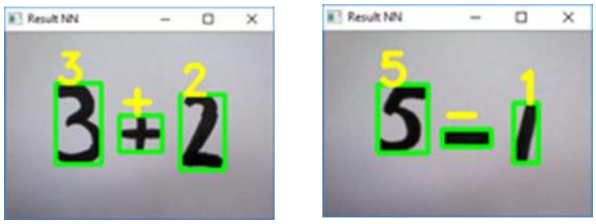

Figure 20. Neural networks recognition

The accuracy for the Support Vector Machine with a Gaussian kernel is 95.15% which makes it the most successful algorithm for this problem among the used classifiers. Random Forest and K-Nearest Neighbours also perform satisfactory with an accuracy rate of 94.37% and 93.72% respectively. However, K-Nearest Neighbours is significantly slower than the other algorithms. The classifier with the lowest accuracy is Stochastic Gradient Descent training of Linear Support Vector Machine with 86.98%, and it runs in a few seconds. (Fig. 21)

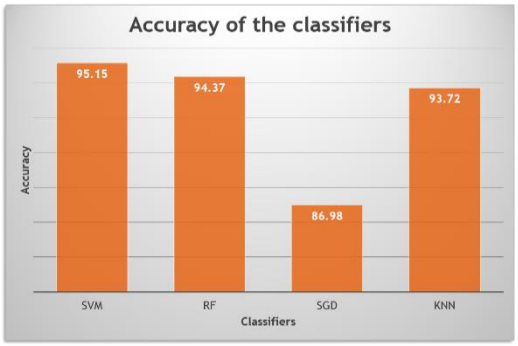

Figure 21. Accuracy of the four classifiers

Testing on actual images with multiple written expressions has been conducted. The total number of images is 30 with one, two or three expressions on each. The expressions contain 1-digit and 2-digit numbers with operators between them. The expressions consist of a total number of characters being 208 which includes all digits 0-9 and the three operators for addition, subtraction and multiplication. The Table I below illustrates the number of true and false recognition with the final accuracy for each of the four classifiers.

TABLE I. ACCURACY OF THE FIVE CLASSIFIERS ON ACTUAL IMAGES

Classifier	True recognition	False recognition	Accuracy
RF	152	56	73.08%
SVM	153	55	73.56%
SGD	133	75	63.94%
KNN	158	50	75.96%
NN	167	41	80.25%

The most successful algorithm on the actual images is Neural Networks with an accuracy of 80.28%. The second successful algorithm is K-Nearest Neighbours with an overall accuracy rate of 75.96%. The third most accurate is the SVM with an accuracy of 73.56% followed by Random Forest with a near accuracy rate of 73.08%. Linear SVM with SGD training expectedly has

the worst performance with 63.94% accuracy which is considerably lower compared to the other three classifiers. (Fig. 22 and Table II)

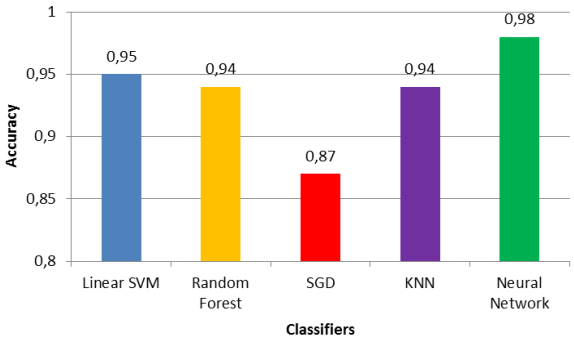

Figure 22. Comparison between the five classifiers

TABLE II. ACCURACY OF THE THREE OBJECT RECOGNITION ALGORITHMS

Type of recognition	Correct cases	Incorrect cases	Total number
Typed text recognition	91	9	100
Colour recognition	89	11	100
Shape and colour recognition	82	18	100

V. DISCUSSION

A. Interpretation of the Results and Possible Improvements

The algorithms for object recognition have been initially tested on images derived from the web or drawn in Paint. The results in these experiments were substantial which allowed me to test the algorithms on images captured by NAO's camera with different lighting conditions, position and rotation. The second type of experiments allowed me to make the needed adjustments to the algorithms in order for them to be applicable for different scenarios. The results for typed text, colour and shapes recognition have achieved an accuracy of above 90%. In terms of typed text, two functions from OpenCV are applied, simple thresholding and adaptive thresholding. As it is already stated, adaptive thresholding produces better results as it manages to recognise text in images where simple thresholding fails. Colour and shape recognition's accuracy is similar to text recognition, and the robot's movements for pointing at the figures can also be regarded as successful. The five machine learning classifiers including artificial neural networks have shown accurate results from the experiments when they are tested on images from the testing dataset from MNIST.

The sign language gestures have been successful, and the robot did not lose balance at any point as only arms are used for the expressions. Due to the fact that the robot has only three fingers instead of five, and they all can be moved at once, prevented from implementing other words in order to enrich the dictionary by making it more comprehensive. Despite the fact that colour recognition produces accurate results, it is limited by the fact that pure red, green and blue are not always recognised as such, and instead, their shades are detected. This can be overcome by replacing the names of the shades with the names of the corresponding primary colours. A drawback of the shapes recognition is the fact that the two figures on the sheet are small in size and have a limited distance between each other, which makes the pointing of the robot at them not so clear. In terms of the machine learning algorithms, future adaptations need to be made for the properties of different pictures such as lighting, rotation and position. The publicly available database MNIST contains digits only, and hence, pluses (+) and minuses (-) were added as part of this research. Thus, the robot is allowed to solve basic mathematical expressions. Moreover, other classifiers can be taken into account which may showcase better results.

B. Ethical Issues Related to the Use of Robots in Educational Institutions

This research project has not been integrated in an educational institution yet, but it provides potential means for successful usage. A considerable amount of research has been conducted in regards to the use of humanoid robots in educational institutions for learning's enhancement. Several studies that introduced NAO robot have been considered in the literature review including the approach that has been used, the benefits and any suggestions of how this could be further improved. Visual learning that is considered in this study is proven to be the most effective way of learning [36]. Any opinions and views of people from different age groups, gender, nationality shared on the web have been taken into consideration in order to evaluate the utility of such use of robots together with any ethical concerns that may arise.

In order for the robot architecture to be acceptable and useful in educational institutions, the accuracy of the object recognition algorithms needs to be significant. This research project provides highly accurate methods bearing in mind the fact that 100% recognition is unachievable. Hence, the system architecture can be successfully tested in educational setting, but further ethical issues need to be considered. In some experiments a colour or a digit can be misclassified as another, which may cause confusion to children. However, as the proposed architecture will be monitored by a teacher, their role will be to correct the error accordingly. Due to the fact that the occurrence of errors is inevitable, the role of the teacher is of high importance as a guide who will need to minimize the use of certain images that seem to cause issues in any lighting and position circumstances.

In terms of ethical concerns, people commonly question the use of robots in schools. As robots are so developed that they take social roles in our society, it is relevant to consider the social, legal and ethical issues raised by such use. It should be defined when it is best to use robots and in what situations they should be avoided. A European survey conducted in 2012 which involved 27,000 people found that 34% thought robots should be banned from educational institutions [37]. There are different social, legal and ethical concerns about robot

teacher and they vary according to the role the robot has and what purpose it is aimed at. In relation to legal issues, they are not currently very well defined in the area of teaching and the major ones include robot testing outside laboratory and any damage that could be caused [38]. The most obvious ethical concern relates to the problem of privacy. This is mainly relevant to situations where the robot collects data from its sensors. The access to this data could be considered as another legal issue. However, it does not apply to this research project as the robot would only capture images with its camera and there would not be any children included on them. A common collection of the ethical concerns addresses the problem about the attachment, deception and loss of human contact. This would not be a major issue in this potential research as human teachers would cooperate with the robots in order to enhance their visual learning. Thus, the children would not lose their human contact since human teachers are meant to cooperate with the robots. A social issue related to this is called 'social valence' and this refers to people's perceptions that the robot may feel like a person to them [39]. The accountability problem is another ethical issue which stresses on the question whether robots should be trusted to make decisions about what the children should do [40]. Again, this is not relevant to this project as the role of the robot would be absent in terms of making decisions about children.

VI. Conclusion and Future Work

The object recognition algorithms for typed text, colour, and shapes have been considerably successful as they are recognised by the robot with accuracy of around 90%. The five machine learning classifiers' accuracy on the testing set varied from 87% to 98%. This project presents a novel approach for children's visual learning enhancement which can be integrated in schools in order to aid the young.

Future work stresses on the improvement of the implemented algorithms in order for them to be applicable for a wider range of images, and this mainly involves the pre-processing of the image prior the use of a function or a classifier. In terms of sign language, the robot can be allowed to perform the reverse operation – it recognises the gesture, and the robot then tells the corresponding word. Further ideas include drawing a figure on a sheet by the robot after it has been recognised, and grasping balls of different colours.

References

[1] A. Liu, J. Newsom, C. Schunn, and R. Shoop, "Students learn programming faster through robotic simulation," *Tech Directions*, 2013.

[2] Y. Amit and P. Felzenszwalb. (2012). Object detection. [Online]. Available: https://cs.brown.edu/~pff/papers/detection.pdf

[3] K. Khurana and R. Awasthi, "Techniques for object recognition in images and multi-object detection," *International Journal of Advanced Research in Computer Engineering & Technology*, vol. 2, no. 4, pp. 1383-1388, 2013.

[4] H. Kurukshetra, "Image processing and object detection," *International Journal of Applied Research*, vol. 1, no. 9, pp. 396-399, 2015.

[5] L. Breiman and A. Cutler. (2014). Statistics at UC Berkeley. *Random Forests*. [Online]. Available: https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home

[6] L. Bottou, *Stochastic Gradient Descent Tricks*, Springer, 2012, pp. 430-445.

[7] S. Gunn, "Support vector machines for classification and regression," University of Southampton, 1998, pp. 5-17.

[8] O. Sutton. (2012). Introduction to k nearest neighbour classification and condensed nearest neighbour data reduction. [Online]. pp. 1-3. Available: http://www.math.le.ac.uk/people/ag153/homepage/KNN/OliverKNN_Talk.pdf

[9] B. Krose and P. V. D. Smagt, *An Introduction to Neural Networks*, UCL Press, 1996, pp. 15-20.

[10] Y. Zamani, Y. Souri, H. Rashidi, and S. Kasaei, "Persian handwritten digit recognition by random forest and convolutional neural networks," in *Proc. 9th Iranian Conference on Machine Vision and Image Processing*, 2015.

[11] H. Khan, "MCS HOG features and SVM based handwritten digit recognition system," *Journal of Intelligent Learning Systems and Applications*, vol. 9, no. 2, pp. 21-33. 2017.

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.

[13] S. Bernard, L. Heutte, and S. Adam, "Using random forests for handwritten digit recognition," in *Proc. IEEE International Conference on Document Analysis and Recognition*, 2007, pp. 1043-1047.

[14] N. Hamid and N. Sjarif. (2017). Handwritten recognition using SVM, KNN and neural network. [Online]. pp. 2-3. Available: https://arxiv.org/ftp/arxiv/papers/1702/1702.00723.pdf

[15] E. Tuba, M. Tuba, and D. Simian, "Handwritten digit recognition by support vector machine optimized by bat algorithm," in *Proc. 24th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2016, pp. 369-376.

[16] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3642–3649.

[17] L. Wan, M. Zeiler, S. Zhang, Y. Lecun, and R. Fergus, "Regularization of neural network using drop connect," in *Proc. International Conference on Machine Learning*, 2013.

[18] V. Romanuke. (2016). Parallel Computing Center (Khmelnitskiy, Ukraine) represents an ensemble of 5 convolutional neural networks which performs on MNIST at 0.21 percent error rate. [Online]. Available: https://www.drive.google.com/file/d/0B1WkCFOvGHDddElkdkl6bzRLRE0/view

[19] C. Wenbai, W. Xibao, W. Sai, and G. Hui, "Human's gesture recognition and imitation based on robot NAO," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 8, no. 12, pp. 259-270, 2015.

[20] D. Albani, A. Youssef, V. Suriani, D. Nardi, and D. Bloisi, "A deep learning approach for object recognition with NAO soccer robots," in *Proc. 20th RoboCup International Symposium*, 2016.

[21] G. Majgaard, "Humanoid robots in the classroom," *IADIS International Journal on WWW/Internet*, vol. 13, no. 1, pp. 72-86, 2017.

[22] H. Altin, A. Aabloo, and G. Anbarjafari, "New era for educational robotics: Replacing teachers with a robotic system to teach alphabet writing," in Proc. *4th International Workshop Teaching Robotics, Teaching with Robotics*, 2014, pp. 164-166.

[23] B. Ertugrul, C. Gurpinar, and H. Kivrak, "Gesture recognition for humanoid assisted interactive sign language tutoring," in *Proc. Signal Processing and Communications Applications Conference*, 2013.

[24] F. Tanaka and S. Matsuzoe, "Children teach a care-receiving robot to promote their learning: Field experiments in a classroom for vocabulary learning," *Journal of Human-Robot Interaction*, pp. 78-95, 2012.

[25] Y. Lecun, C. Cortes, and C. Burges. (2017). MNIST handwritten digit database, YannLeCun, Corinna Cortes and Chris Burges. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[26] Scikit-image development team. (2017). Histogram of Oriented Gradients — skimage v0.14dev docs. [Online]. Available:

http://scikitimage.org/docs/dev/auto_examples/features_detection/plot_hog.html
[27] Scikit-learn. (2017). 1.13. Feature selection — scikit-learn 0.19.1 documentation. [Online]. Available: http://scikit-learn.org/stable/modules/feature_selection.html
[28] D. Comaniciu and P. Meer, "Robust analysis of feature spaces: Color image segmentation," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997, pp. 1-5.
[29] Scikit-learn. (2017). 4.2. Feature extraction — scikit-learn 0.19.1 documentation. [Online]. Available: http://scikit-learn.org/stable/modules/feature_extraction.html
[30] A. Rosebrock, *Practical Python and OpenCV*, 3rd ed., 2016, pp. 113-114.
[31] E. Turkel. (2012). *Automatic Thresholding*. [Online]. Available: http://www.math.tau.ac.il/~turkel/notes/otsu.pdf
[32] J. Forcier. (2017). Welcome to Paramiko's documentation! — Paramiko documentation. [Online]. Available: http://docs.paramiko.org/en/2.1/
[33] OpenCV documentation. (2017). *OpenCV: Image Thresholding*. [Online]. Available: https://docs.opencv.org/3.3.1/d7/d4d/tutorial_py_thresholding.html
[34] R. Kohavi and F. Provost, "Glossary of terms," *Applications of Machine Learning and the Knowledge Discovery Process*, vol. 30, pp. 271-274, 1998.
[35] G. Piñeiro, S. Perelman, J. Guerschman, and J. Paruelo, "How to evaluate models: Observed vs. predicted or predicted vs. observed?" *Ecological Modelling*, vol. 216, no. 3-4, pp. 316-322, 2008.
[36] H. Kouyoumdjian. (2012). *Learning Through Visuals*. [Online]. Available: https://www.psychologytoday.com/blog/get-psyched/201207/learning-through-visuals
[37] Eurobarometer 382. (2012). Public attitudes towards robots. Bussels: European Commission. [Online]. Available: http://cordis.europa.eu/fp7/ict/robotics/docs/special-eb-survey-382-public-attitudes-towards-robots-summary_en.pdf
[38] A. Bertolini, *et al*. (2017). *The legal issues of robotics* | Robohub. [Online]. Robohub.org. Available: http://robohub.org/the-legal-issues-of-robotics/
[39] C. McDonald. (2015). The good, the bad and the robot: Experts are trying to make machines be "moral". Cal Alumni Association. [Online]. Available: https://alumni.berkeley.edu/california-magazine/just-in/2015-06-08/good-bad-and-robotexperts-are-trying-make-machines-be-moral
[40] A. Sharkey. (2016). Should we welcome robot teachers? [Online]. Available: https://link.springer.com/article/10.1007/s10676-016-9387-z

Yordanka Karayaneva was born in Bulgaria. She received her Bachelor of Science degree in Computer Science in 2017 from Coventry University, United Kingdom. Currently, she is a PhD student at the same institution in robotics and computer science. Karayaneva's PhD research is based on developing a holistic methodology for determining applications for robotics in elderly care homes.
She undertook two summer internships in robotics and reinforcement learning in June-July 2016, and security in June-August 2017 at Coventry University. Her research interests include machine learning, computer vision, object recognition and human-computer interaction.

Dr. Diana Hintea is a faculty member of the School of Computing, Electronics and Mathematics at Coventry University, UK. She received her PhD in Computer Science from Coventry University in 2015 and is now the Course Director for the Computer Science course at Coventry University.
Her research areas are machine learning, artificial intelligence, data analysis and digital forensics.