

Received April 17, 2020, accepted May 3, 2020, date of publication May 6, 2020, date of current version May 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2992614

A Deep Learning-Based Pipeline for Teaching Control Theory: Transforming Feedback Control Systems on Whiteboard Into MATLAB

**DORUKHAN ERDEM, AYKUT BEKE^{ID}, (Student Member, IEEE),
AND TUFAN KUMBASAR^{ID}, (Senior Member, IEEE)**

Department of Control and Automation Engineering, Istanbul Technical University, TR-34469 Istanbul, Turkey

Corresponding author: Tufan Kumbasar (kumbasart@itu.edu.tr)

This work was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under Project 118E807.

ABSTRACT The difficulty in teaching control theory is that the lecturer must not only provide all the theoretical concepts but also visualize the control system in time and frequency domains. In control system courses, the visualizations are usually provided with roughly sketches on whiteboards and thus might be difficult to understand. In this paper, a Deep Learning (DL) based pipeline is proposed that is capable to recognize Handwritten Feedback Control Architectures (HFCAs) on the whiteboard and to transform them into Matlab[®] for visualization and analysis of control systems interactively. The proposed DL pipeline consists of 5 main steps that take up the frameworks of deep learning, pattern recognition and image processing. The main challenges of constructing such a pipeline are the uncertainties resulting from the lecturer's handwriting quality and lighting conditions in the classroom, which can be seen as inter- and intra-quality uncertainties. Therefore, we employed and trained deep Convolutional Neural Networks (CNNs) to recognize the HFCAs with a high performance. In the training of deep CNNs, we integrated the transfer learning approach with the deep CNN ResNet-50. To capture the inter- and intra- quality uncertainties, we constructed an image dataset of HFCAs collected from control system lecturers, who have different levels of experience, in a small-sized classroom under different lighting conditions. We provide all the details on the design of the DL based pipeline and present experimental results to show that the pipeline is a powerful tool to visualize HFCAs in real-time by using the advantages of Matlab[®].

INDEX TERMS Control theory, deep learning, image processing, MATLAB, pattern recognition, ResNet-50, visualization.

I. INTRODUCTION

Teaching control theory is difficult as there are many theoretical concepts to be addressed. The main difficulty that students face is visualizing and understanding the relationship between the time and frequency domain parameters of a control system [1]–[3]. Therefore, the visualization of control systems is crucial to demonstrate the role of mathematics in control system design [1]. As this problem is not new, various approaches have been proposed to provide innovative techniques to enhance the students' motivation and improve their comprehension of control theory. For instance, interactive software tools are presented for teaching control systems

in [4]–[8]. Besides, remote & virtual control laboratories are developed to provide students with a hands-on experience of control systems [9]–[12].

In most of the control system design courses, the main focus is usually on Feedback Control Architectures (FCAs) that are composed of controllers and Transfer Functions (TFs) structured within single/multi loop configurations [13]. In Fig. 1, the most commonly handled FCAs are shown and their descriptions are provided in Table 1. The teaching approach of control system design is usually performed in a three-fold approach. Firstly, the lecturer defines one of the FCA (shown in Fig. 1) and then analyses it in the time and/or frequency domain. Finally, the lecturer provides the students with theoretical background on controller design approaches such as graphical (Bode and Root-Locus plots) or automatic

The associate editor coordinating the review of this manuscript and approving it for publication was Jesus Felez^{ID}.

TABLE 1. Descriptions of FCAs defined in [14].

FCA	Descriptions
FCA-1	<ul style="list-style-type: none"> • Compensator $C(s)$ and plant $G(s)$ in forward path • Sensor dynamics $H(s)$ in feedback path • Prefilter $F(s)$
FCA-2	<ul style="list-style-type: none"> • Single feedback loop • Plant $G(s)$ in forward path • Compensator $C(s)$ and sensor dynamics $H(s)$ in feedback path, • Prefilter $F(s)$
FCA-3	<ul style="list-style-type: none"> • Compensator $C(s)$ and plant $G(s)$ in forward path • Sensor dynamics $H(s)$ in feedback path • Feedforward prefilter $F(s)$ for input disturbance attenuation.
FCA-4	<ul style="list-style-type: none"> • Outer loop with compensator $C1(s)$ in forward path • Inner loop with compensator $C2(s)$ in feedback path • Plant $G(s)$ in forward path • Sensor dynamics $H(s)$ in feedback path
FCA-5	<ul style="list-style-type: none"> • Compensator $C(s)$ in forward path • Plant $G1(s)$ and plant predictive model $G2(s)$ • Prefilter $F(s)$
FCA-6	<ul style="list-style-type: none"> • Plant models $G1(s)$ and $G2(s)$, compensators $C1(s)$ and $C2(s)$ in the forward path • Sensor dynamics $H1(s)$ and $H2(s)$ in the feedback path of both loops. • Prefilter $F(s)$

(LQR and IMC tuning) tuning methods [13]. To design and analyze FCAs, Matlab[®] provides an excellent environment; especially its Simulink[™] and the Control System Toolbox[™][14]. The FCAs that are shown in Fig. 1 can be easily analyzed and designed via the user interface of the Control System Designer[™] as they built-in structures.

Teaching control system design is usually performed in an old-fashion style with a whiteboard. The lecturer basically defines one of the FCAs on the whiteboard as shown in Fig. 2. Although whiteboards are easy to use, we believe that this results with the following bottlenecks in teaching control:

- The visualization of the control system (in the frequency and time domain) is poor since it is (almost always) roughly sketched and not scaled as it can be seen from Fig. 2. The visualization is even sometimes impossible for high-order control systems.
- The sensitivity to system parameter variations and controller parameters is hard to illustrate during lecture in real-time.
- The reference tracking and disturbance rejection performances are hard to show during lecture in real-time.
- The graphical and automatic tuning methods cannot be directly employed in real-time.

Besides, the lighting conditions in the classroom (as it can be seen from Fig. 2) and more importantly the lecturer's handwriting quality result with difficulties in grasping the concepts of control theory since the students have to put an extra effort to understand/ recognize the handwritings on the whiteboard. These issues can be seen as uncertainties that can be categorized in following forms:

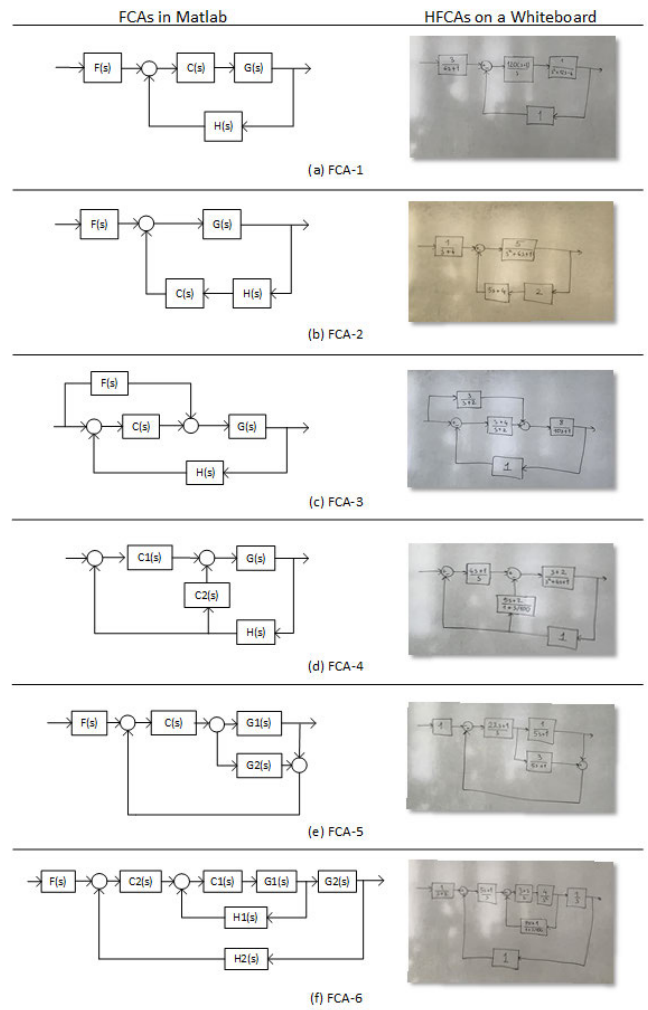


FIGURE 1. Illustration of the FCAs.

- **Inter-quality uncertainty:** The variation amongst the quality of the lecturers' handwritings. The quality of the lecturer depends heavily on the experience of the lecturer in writing on the whiteboard.
- **Intra-quality uncertainty:** The variation of the quality of the lecturers' handwriting over time. The quality of a lecturer's handwriting might degrade over a three-hour course due to tiredness. Moreover, the lighting conditions might change in that period and thus cause reflections on the board.

Similar versions of the aforementioned uncertainties have been widely encountered in handwritten flowchart and character recognition [15]–[17]. In the handwritten diagram detection literature, most of the studies deal with ink-input devices and use the spatial information of pen strokes in their recognition method [17]–[21]. There are also a few studies using image processing techniques to recognize handwritten diagrams [22]–[24].

In the last decade, we have witnessed that Deep Learning (DL) based pattern recognition systems have been successfully employed in various areas [25]–[27]. In the

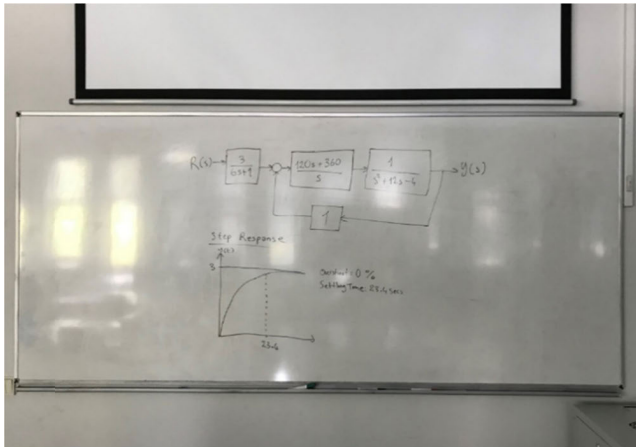


FIGURE 2. Illustration of a HFCA with a poor time domain visualization on a whiteboard.

literature, Deep Convolutional Neural Networks (CNNs) have been usually employed in this context since they can extract refined features from a low-level input image/vector and thus can successfully recognize the object in the image. The deployment of Deep CNNs demonstrated significant performance improvements in pattern recognition such as in character recognition [27]–[29], document recognition [30], face recognition [31] and image recognition [32].

In this paper, we proposed a DL based pipeline that is capable to recognize Handwritten FCAs (HFCAs) on the whiteboard and to transform them into Matlab[®] for visualization and analysis of FCAs. The main design challenge of the proposed DL based pipeline are the aforementioned uncertainties, namely Inter- and Intra- quality uncertainties. In this context, the pipeline takes up and uses the frameworks of DL, pattern recognition and image processing to provide an efficient solution. In the pipeline, we used and employed DL methods to successfully recognize the HFCAs and the handwritten characters. We preferred the transfer learning approach to construct deep CNNs based on ResNet-50 with datasets constructed from five control system lecturers. The proposed DL based pipeline takes the HFCA (shown in the right of Fig.1) as an input and recognizes the FCA class (shown in the left of Fig.1) via a deep CNN structure. Then, the recognized HFCA is further processed to extract the TF blocks by using image processing methods. The characters of the extracted TFs are recognized and labeled with another deep CNN structure. After generating symbolic expressions, continuous time TF representations are generated that are compatible with Matlab[®]. The visualization and analysis of the HFCA is then straightforwardly performed via the Control System Design[™]Toolbox and Simulink of Matlab[®] in real-time. We present various results to show the efficiency of the proposed DL based pipeline. The main features of the proposed DL based pipeline are:

- The novel DL based pipeline that is capable to recognize HFCAs on the whiteboard and transform them into Matlab[®] in real-time.

- The proposed DL based pipeline is a powerful tool to visualize HFCAs in the time and frequency domain as it uses the advantages of Matlab[®].
- The inter-quality uncertainty, the variation amongst the quality of the lecturers' handwriting, and intra-quality uncertainty, the variation of the quality of the lecturers' handwritings over time and lighting conditions, are handled and captured with deep CNNs.
- The proposed DL based pipeline is a powerful tool to analyze HFCA as it can illustrate the effects of parameter and structure updates/changes in real time when the HFCA is modified.
- Lecturers can use the developed DL based pipeline as complement to their control system courses as it is capable to provide visualizations of control systems in real-time.

The paper organized as follows. Section II presents information on deep CNNs and the integration of transfer learning with ResNet-50. Section III presents the proposed DL based pipeline and provides detailed information and explanation of its components. Section IV presents the observations driven from the real-time experiments. Finally, conclusion and future works are given in Section V.

II. DEEP CONVOLUTIONAL NEURAL NETWORKS

We firstly provide a brief information on CNNs for classification tasks. Then, we present the integration of transfer learning with ResNet-50.

A. BRIEF OVERVIEW ON CNNs

CNNs are feedforward networks which consist of mostly convolutional layers and pooling layers with one or more fully connected layers at the end [33]. A simple CNN structure is illustrated in Fig. 3. On the other hand, Deep CNN architectures are constructed by stacking large number of convolutional, pooling and fully connected layers together.

The convolutional layers can be seen as feature extractors since they learn feature representations of their inputs. Features are extracted by convolving inputs with the learnable weights of the layer. In its learning progress, a convolutional layer updates its weights to select more descriptive features. The convolved results are pass through an activation function such as the Rectified Linear Unit (ReLU) [27], [34]. The output of the k^{th} convolutional layer y_k is as follows:

$$y_k = f(y_{k-1} * w_k) \quad (1)$$

where w_k represents the weights of the k^{th} layer, f represents the activation function, $*$ denotes the 2D convolution operation and y_{k-1} is the previous layer's output. If the k^{th} layer is the first convolutional layer, then y_{k-1} is the input image x . More detail on how convolution operation is conducted in convolutional layers can be found in [34].

Pooling layers act as the downsampler in the CNN. They are usually placed between consequent convolutional layers in order to reduce the dimensions of the input feature maps without introducing new learnable parameters. By doing so,

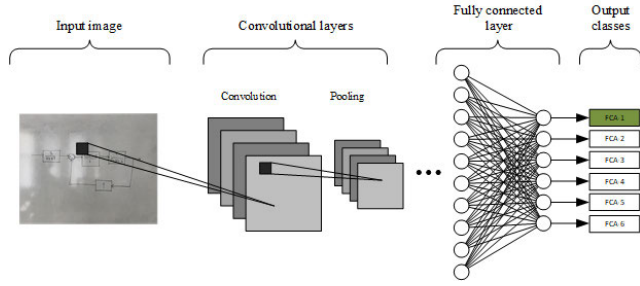


FIGURE 3. Illustration of a shallow CNN architecture.

the network becomes more invariant to small changes in the input images [30]. Pooling layers downsamples the feature maps by selecting local regions from its input and returning a single element for each region. The pooling operation can be accomplished with max or average pooling layers [33].

The fully connected layers in CNNs flatten the outputs of the previous layers and then applies weights to predict the correct labels. The final fully connected layer in classification tasks provides probabilities for each class. It is standard to use the softmax operator to classify the image to a label [35].

B. RESNET-50 AND TRANSFER LEARNING

In our proposed pipeline, we designed deep CNNs by integrating transfer learning approach with ResNet-50.

ResNet-50 is a pre-trained deep CNN architecture which is trained for object detection task on the ImageNet dataset [36]. ResNet-50 has 50 layers with a total number of 23.6 million trainable parameters. What differs ResNet-50 from other CNNs is that it is constructed with a series of bottleneck blocks consisting of stacked convolutional layers with skip connections where a Batch Normalization (BN) and ReLU follows each convolutional layer as illustrated in Fig. 4 [36].

For new pattern recognition tasks, the transfer learning approach is widely employed as training of a deep CNNs, like ResNet-50, from scratch is computational expensive and time-consuming [34], [37]. We would like to underline that the transfer learning approach of ResNet-50 gives the opportunity to use the pre-trained feature extraction layers and fine-tune the ResNet-50 with a smaller size dataset that contains patterns of our handled multi-class recognition problem.

In this study, in transfer learning of ResNet-50, only the classification layer of the ResNet-50 network is replaced and learned. Let us define a classification layer as follows:

$$O = \text{softmax}(w_f \cdot X + b_f) \tag{2}$$

where X is the output of the second to last layer of ResNet-50 and $w = [w_f, b_f]$ defines the learnable parameters, namely weight and bias. The output size of the classification layer is equal to the total number of classes (Q). We have used the cross entropy as the loss function with L2 regularization that is defined as:

$$L = -\frac{1}{Q} \sum_{q=1}^Q [y_q \log \hat{y}_q + (1 - y_q) \log (1 - \hat{y}_q)] + \lambda \frac{w^T w}{2} \tag{3}$$

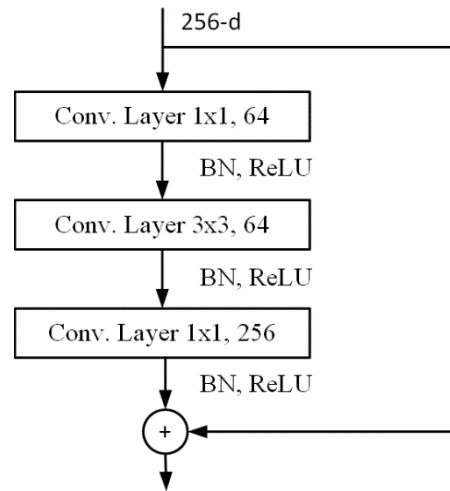


FIGURE 4. A block of a bottleneck: An example.

where λ is the regularization term, \hat{y}_q is the predicted label and y_q is the target label. We preferred to use the Stochastic Gradient Descent with Momentum (SGDM) as the optimizer for the sake of its simplicity and efficiency in training time [34].

III. THE DL BASED PIPELINE

In this section, we present the structure and steps of the DL based the pipeline. An overview of the DL based pipeline is given in Fig. 5 that is summarized with following steps:

- Step-1:** Recognizing the structure of the HFCA with DL.
- Step-2:** Detecting TF blocks using image processing.
- Step-3:** Segmenting and recognizing the characters with DL.
- Step-4:** Constructing symbolic expressions from the recognized characters to construct continuous time TFs in Matlab®.
- Step-5:** Generating the recognized continuous time HFCA in in Matlab®.

In **Step-1** and **Step-3** of the proposed DL based pipeline, the following two pattern recognition problems are defined.

- HFCA Recognition (HFCAR)
- Handwritten Character Recognition (HCR)

In HFCAR and HCR problems, as it can be seen from Fig. 1 and Fig. 2, the main challenge arises from the quality of the lecturer’s handwriting and lighting conditions. To handle such uncertainties, i.e. to model the inter- and intra-quality uncertainty, we used the transfer learning approach of ResNet-50 and constructed deep CNN as described in Section II.B.

In order to train the deep CNN, we constructed an image dataset collected from lecturers of control systems courses in a small sized classroom environment in the presence of different lighting conditions. We collected HFCA from five lecturers that work in the Department of Control and Automation Engineering, Istanbul Technical University and have different

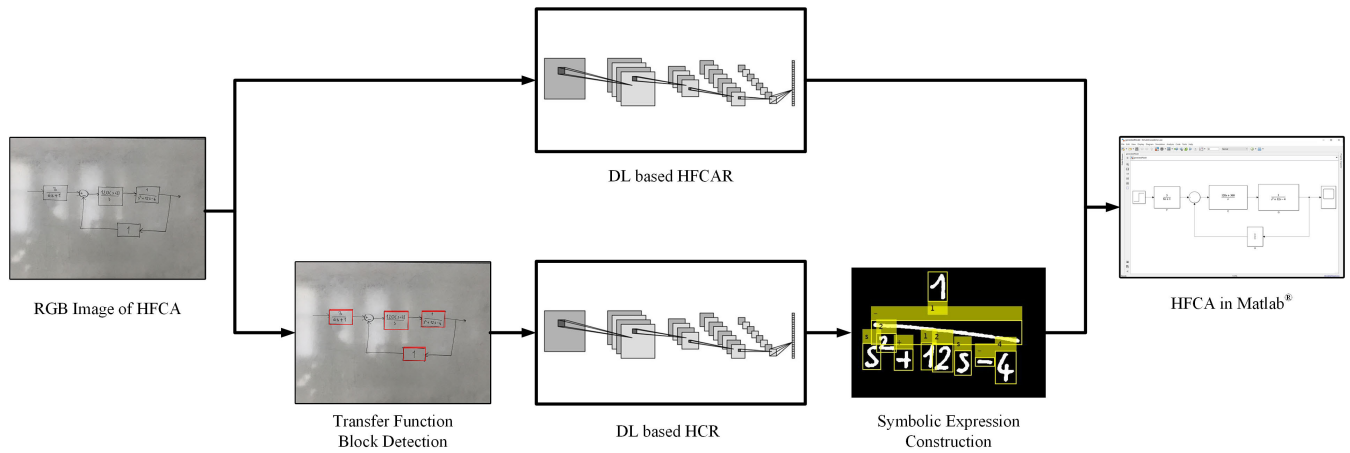


FIGURE 5. Overview of the proposed DL pipeline.

levels of experience in teaching. One lecturer is an Assistant Professor with an experiences of more than 15 years, one of them is an Associate Professor with an experience of more than 10 years, one of them is a Senior Researcher with an experience of more than 3 years while two of them are Young Researchers with experiences of less than 2 years in teaching control systems. We asked each lecturer to draw the each FCA at least 10 times during different times of the day (i.e. different lighting conditions) and also let them to define the TFs within the each FCAs.

The DL based pipeline is implemented in Matlab and CUDA environments on a PC that includes Intel Core i7 3.3GHz CPU, 32GB RAM and NVIDIA GTX 1080 TI GPU. The implementation of DL has been done with the Deep Learning ToolboxTM to have an easy integration with Control System ToolboxTM and SimulinkTM of Matlab[®].

In the latter, we provide detailed description of the steps of DL based pipeline shown in Fig. 5. For illustrative purposes, the steps of pipeline are illustrated on an example HFCA which is shown in Fig. 6.

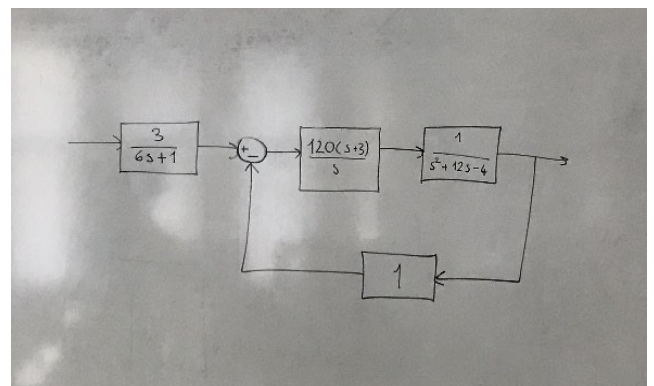


FIGURE 6. An example HFCA (i.e. FCA-1).

TABLE 2. Performance of deep CNN for HFCA.

	Best Accuracy	Mean Accuracy
Training	100%	100%
Validation	97.2%	94.4%
Testing	94.4%	89.25%

A. HANDWRITTEN FEEDBACK CONTROL ARCHITECTURE RECOGNITION WITH DEEP LEARNING

The first task to be accomplished in the proposed DL based pipeline is to solve the HFCAR problem to identify one of the FCAs shown in Fig. 1 (i.e. $Q = 6$ classes). We constructed a HFCA dataset with 306 RGB images with a resolution of 4032×3024 which were captured from an actual whiteboard. The dataset is labeled manually with the classes and is then split as 216 images (36 per class) for training, 36 images (6 per class) for validation and 54 images (9 per class) for testing.

The multi-label HFCAR problem is solved with a ResNet-50 based CNN as described in Section II.B. In learning of the deep CNN, all images are resized to a resolution of 224×224 without any further pre-processing. Furthermore, we employed the online data augmentation method to create

artificially modified versions of images to increase the size of training dataset by slightly rotating and scaling each image at each training epoch. The DL hyperparameters are set as: 100 epochs, minibatch size of 1, and a learning rate of 10^{-3} with a drop rate factor of 0.1 at every 10 epochs.

The best and mean training, validation and testing accuracies over 5 experiments are tabulated in Table 2. The mean training and validation accuracy values are given in Fig. 7 (only the first 3000 iterations are given). It can be concluded that the performance of the deep CNN is satisfactory as it resulted with mean testing accuracy value of 89.25%.

B. TRANSFER FUNCTION BLOCK DETECTION

After recognizing the class of the HFCA, we extracted the TF blocks in the image that are enclosed with rectangular shapes. We used the following rectangle detection algorithm to obtain

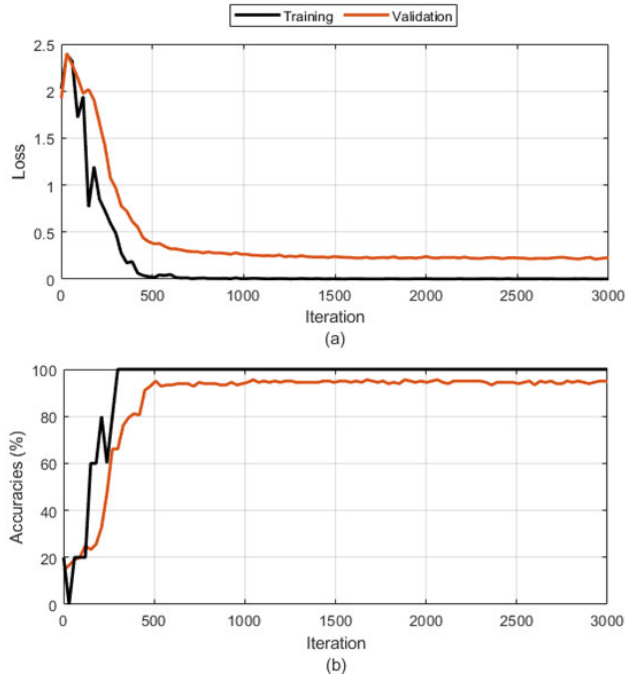


FIGURE 7. HFCAR mean (a) loss values (b) accuracy values.

the location and dimension information of the TF blocks in the HFCA.

- **Binarization:** In order to deal with faded fonts and light reflections like in Fig. 6, a convolution operation is employed with the following n by n kernel k_{ij} :

$$k_{ij} = \begin{cases} 1/n & i = \frac{n-1}{2}, j = \frac{n-1}{2} \\ -1 & \text{else} \end{cases} \quad (4)$$

If a pixel's intensity value is darker when compared to its n by n neighborhood, the result of the convolution operation has a positive value. We defined a positive threshold value D to determine the binary version of the input image. By trial and error, we found that a 15×15 kernel with $D = 15$ is suitable value for our purposes. The binary version of Fig. 6 is given in Fig. 8a.

- **Character and noise removal:** The edges of the FCA in Fig. 8a are then obtained by finding the biggest connected component with 8 connectivity (i.e. connected component with highest pixel count). All remaining components, which are noise and/or character related, are removed from the binary image. The resulting segmentation of the HFCA image is shown in Fig. 8b. Note that it is very likely that the lecturer might not connect all the blocks perfectly. Thus, a dilation operation was also applied to close such disconnections.
- **Filling:** To find the regions in Fig. 8b, the canny edge algorithm is deployed to remove the outermost edge loop that wraps the whole diagram. Then, a morphological fill algorithm is employed to the remaining image and then

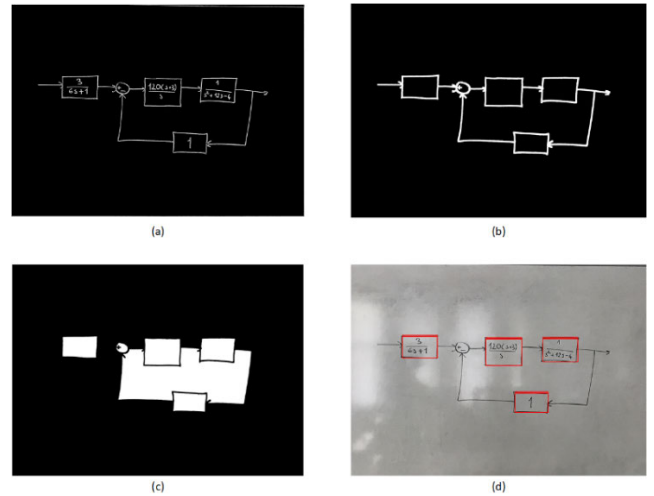


FIGURE 8. (a) Binary image (b) Character, noise removed image (c) Filled image (d) Final image.

the regions that are candidate TF blocks are extracted. Result of filling operation is given in Fig. 8c.

- **Rectangle extraction:** The rectangular shapes are extracted by employing connected component analysis to the filled image. Then, we perform shape analysis by calculating the circularity and rectangularity of each component. The Circularity measure (C) is defined as:

$$C = \frac{4\pi A}{P^2} \quad (5)$$

where A is the area and P is the perimeter of the region in terms of pixels. The Rectangularity measure (R) is defined as:

$$R = \frac{A}{L_{major}L_{minor}} \quad (6)$$

where L_{major} is the length of the major axis and L_{minor} is the length of the minor axis of the region. We defined a connected component as a rectangular shape that satisfies the following criteria:

$$\begin{aligned} C &\leq 0.9 \\ 0.7 &\leq R \leq 1.5 \end{aligned} \quad (7)$$

In Fig. 8d, we show the connected components that satisfy the criteria which are labeled as TF blocks.

C. HANDWRITTEN CHARACTER RECOGNITION WITH DEEP LEARNING

The characters inside the extracted TF blocks are firstly segmented and then recognized. The input image is cropped by using the extracted Region Of Interest (ROI) and thus the resulting image inherits only the characters inside the extracted TF block as shown in Fig. 9a. Then, to find each character, a segmentation method using the connected components analysis is performed as shown in Fig. 9b. Afterward, the characters are cropped and reshaped into images with 224×224 resolution.

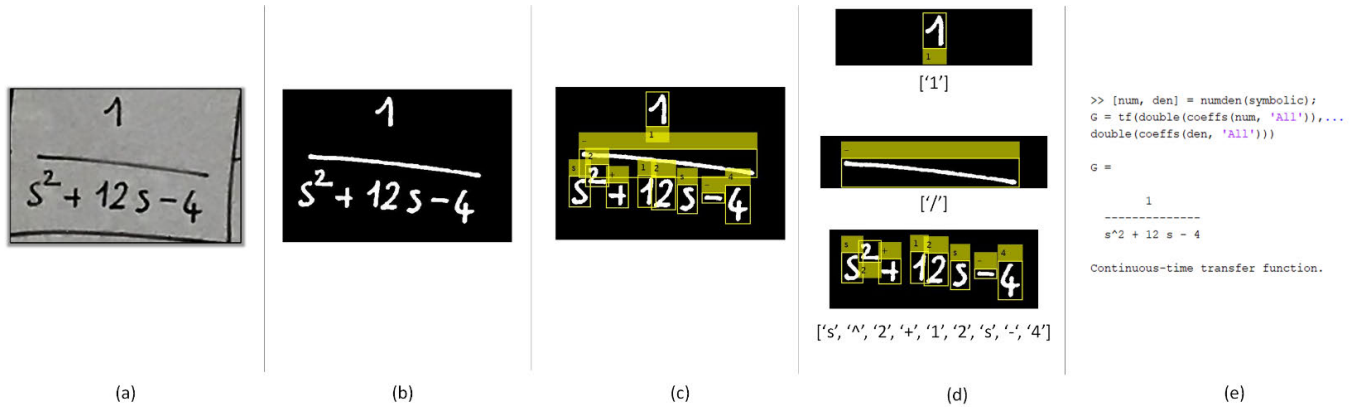


FIGURE 9. (a) extracted TF block image (b) segmented TF block image (c) labelled TF block image (d) labeled numerator and denominator (e) TF in Matlab.

TABLE 3. Performance of the deep CNN for HCR.

	Best Accuracy	Mean Accuracy
Training	100%	100%
Validation	98.39%	97.2%
Testing	96.17%	96.08%

The characters to be recognized are digits (0-9), arithmetic operators ('+', '-', 'x', '*'), round and square bracket pairs, and the 's' and '.' characters, which makes in total $Q = 20$ classes. In the dataset construction for HCR, we used the whole constructed HFCA dataset containing 306 samples by first extracting TF blocks then segmenting characters via the aforementioned approach. It is worth to underline that we have observed that lighting conditions had much bigger impact on small images and therefore we decided to extract the characters from the binary version of the HFCA images.

Each extracted character is labelled manually by the authors. An example of the used character images is shown in Fig. 10. Moreover, we would like to point out that the dataset is enriched with extra handwritten character images collected from the lecturers to end up with evenly distributed samples for each class. The HCR dataset has 3920 images in total which is split as 155 images per class for training, 32 images per class for validation and 9 images per class for testing.

The HCR problem is solved with a deep CNN that is trained as described in Section II.B with hyperparameter settings of 50 epochs, minibatch size of 4, and a learning rate of 10^{-3} with a drop rate factor of 0.1 at every 10 epochs. We also employed online data augmentation in the learning.

The best and mean training, validation and testing accuracies over 5 experiments are given in Table 3. The mean training and validation accuracy values are illustrated in Fig. 11 (only the first 500 iterations are given). It can be seen that the learning performance of the ResNet-50 based deep CNN is satisfactory since it resulted with mean accuracy of more than 96%. As it can be seen from Fig. 9c, the trained deep CNN is capable to successfully label the characters of the segmented TF image.

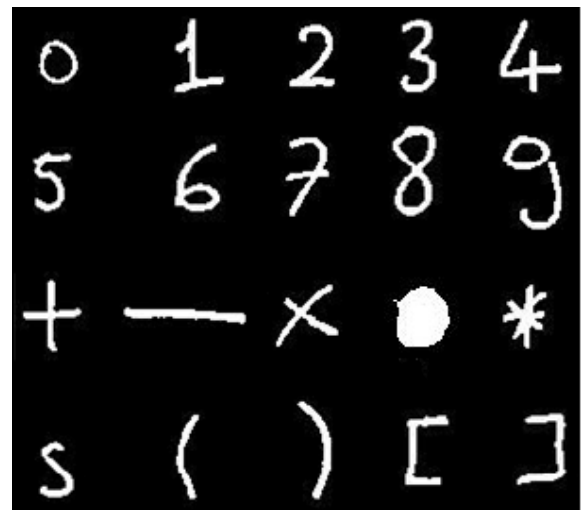


FIGURE 10. Example character images.

D. SYMBOLIC EXPRESSION CONSTRUCTION

We present firstly how we constructed the symbolic expression from the labeled images. Then, by extracting the coefficients of these symbolic expression, we convert them to representations that are compatible with Matlab®.

A single input-single output TF is defined as a ratio of two polynomials, i.e. numerator and denominator. To find the coefficients of these polynomials, we first build an equation string using the recognized characters as shown in Fig. 9c, and then construct a symbolic expression from that string.

A visual summary of symbolic expression construction process is given in Fig. 9d to Fig. 9e. To distinguish the numerator and denominator polynomials of the recognized TF, the characters labeled with the class "-" are firstly examined. To differentiate whether this label represents the subtraction or fraction operator, we simply checked if there is another labeled character above and below of its position. If this condition results in a non-empty set, then we concluded that the labeled character is a fraction symbol that separates the numerator and denominator of TF as shown in Fig. 9d.

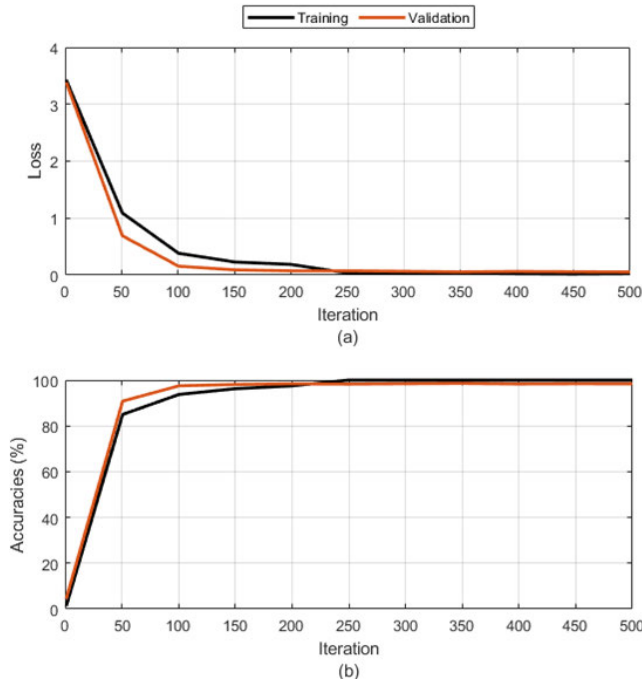


FIGURE 11. HCR mean (a) loss values (b) accuracy values.

Then, the remaining characters are allocated as elements of the sets defining the numerator and denominator part of TF with respect to their position. In this step, we have also handled exponent characters in the polynomials. A character is labeled as an exponent if it is positioned (slightly) above of its preceding character. As shown in Fig. 9d, the first character labeled as “2” is an exponent, because of its relative vertical position to the first occurring “s” character. Thus, we add a caret character (^) between the base and the exponent characters.

Once all the characters are allocated in order, the character sets of the numerators and denominators are turned into strings and then are merged with a division symbol to obtain a string expression of the TF image. This string expression is then transformed into a symbolic expression to define TF representation in which the “s” character becomes the only symbolic variable. Finally, the coefficients of the symbolic expressions of the numerator and denominator are extracted to define TFs in Matlab[®].

E. FEEDBACK CONTROL ARCHITECTURE GENERATION IN MATLAB[®]

Here, we explain how we generate the recognized HFCA in Matlab[®]. Once the deep CNN trained to solve HFCAR problem recognizes the FCA class, the extracted TF representations of the image have to be matched with their appropriate slots in the FCA. To accomplish such a goal, we firstly assign the extracted TFs that have similar vertical positions in the image to the same path using the center coordinates of the extracted TF blocks that are calculated via their ROI information. Then we name and match them with the corresponding TFs (such as G(s), C(s), H(s)...) defined in the FCAs through

their horizontal positions in the image. Note that, we defined a path as a horizontal route a signal can follow in FCAs (i.e. feedforward or feedback path).

In order to provide a clear understanding, let us explain the matching and naming of the TFs on the FCA-1 structure for illustrative purposes. As it can be observed from Fig. 1a, the FCA-1 has two paths including a feedforward path with 3 TFs (F(s), C(s) and G(s)) at top and a feedback path with a single TF (H(s)). If a HFCA is recognized as FCA-1 and all the TFs are extracted in the image frame, then we end up with 3 TFs aligned and a single TF near the upper and lower half of the image, respectively. In the FCA-1, since we know that the prefilter F(s) is the first TF in the feedforward path, we name and match the extracted TF with smaller horizontal coordinate at the upper path as the TF F(s), while the next one to its right as the compensator TF C(s) and the rightmost one as the plant TF G(s). The remaining extracted TF is directly matched and named as the TF that defines the sensor dynamics H(s) since the feedback path of FCA-1 contains a single TF. In a similar manner, the rest 8 FCAs are matched with the extracted TFs in proposed DL based pipeline. Now, the FCA constructed from HFCA can be directly processed and analyzed in Matlab[®] since all the TFs of FCA are defined in the workspace of Matlab[®].

In order to analyze the FCAs via the Control System Designer App[™] of Matlab[®], we define a Matlab object in which the recognized FCA class is defined with the extracted and matched TFs and then import it to the graphical user interface of the application. As shown in Fig. 12, the user can now not only visualize the control system in the time and frequency domains but also tune the compensator C(s).

It is also worth to mention that the DL based pipeline automatically generates a Simulink[™] diagram as shown in Fig. 12d which can be directly used for simulation purposes. To accomplish such a goal, we created template Simulink files in advance for all the FCAs in which all TFs are named as defined in Table 1. In the template Simulink files, the simulation time and solver options are defined with the default settings of Simulink[™]. Once the HFCA is recognized, the corresponding Simulink file is automatically opened and the matched TFs are loaded into the file.

IV. REAL-TIME PERFORMANCE OF THE DL BASED PIPELINE

We performed a series of experiments in order to evaluate the performance of the DL based pipeline in a small-sized classroom. *The real-time performance of the DL based pipeline can be observed from the video file provided as a multimedia file to this paper.* We can conclude from the provided results that the DL based pipeline

- can handle Inter-quality uncertainty as it is capable to recognize FCAs from different users in real-time.
- can handle Intra-quality uncertainty as it is capable to recognize FCAs from in the presence of various lighting conditions in real-time.

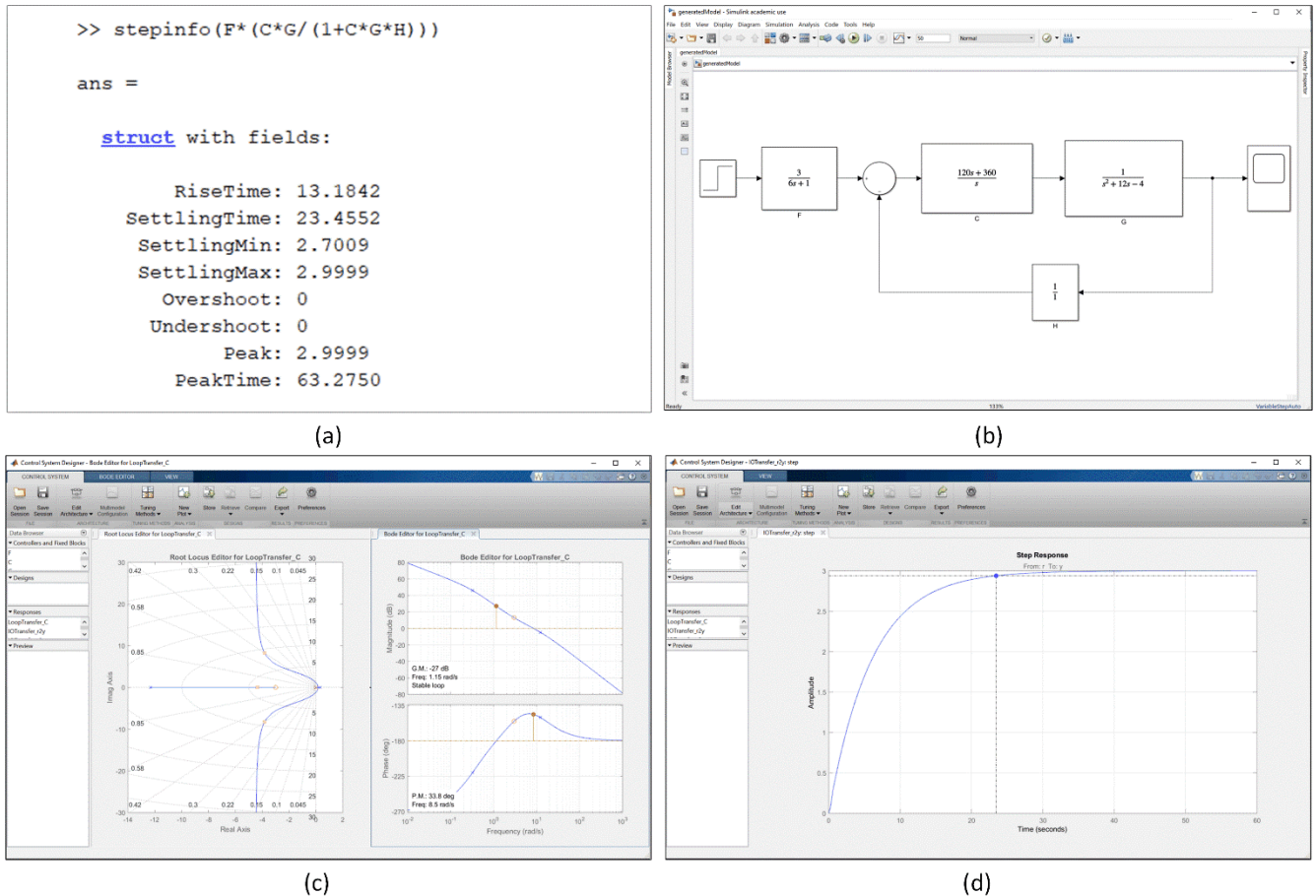


FIGURE 12. (a) Matlab command window (b) Simulink diagram (c) Root-locus and Bode plots (d) Step response.

- is an efficient tool to visualize HFCAs in the time and frequency domain as it uses the advantages of Matlab®.
- is a powerful tool to analyze HFCA as it can update the structure and parameters in real time when there is a change in HFCA.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we provided a DL based pipeline that provides the opportunity to the lecturers/ researchers to visualize and analyze HFCAs during a lecture by transforming the HFCA into Matlab® in real-time. To accomplish such a goal, the frameworks of deep learning, pattern recognition and image processing have been integrated in a novel pipeline. We provided all the details necessary information to construct the proposed DL based pipeline. In the pipeline, we integrated deep CNNs to solve the pattern recognition problems, HCAR and HCR, in order to capture and handle the intra-quality and inter-quality uncertainties that mainly occur due to handwriting quality of the lecturers and lighting conditions. We provided real-time experimental results conducted in a small-sized classroom and clearly showed that the DL based pipeline is a powerful tool to recognize HFCAs and to visualize as it directly uses the advantages of Matlab®. It is also worth to underline that the developed DL based pipeline

is capable to transform the HFCAs on the whiteboard into Matlab® in real-time and thus is capable to update the FCA generated in Matlab® if there is a modification in the HFCA.

We think that the DL based pipeline has the potential to ease the difficulty in teaching control systems as real-time visualizations of control systems and simulations are generated as the lecturer is sketching FCAs during the lectures. As for our future work, we plan the inclusion of discrete-time HFCAs, only the character “z” has to be included as a label in HCR problem, and continuous and discrete time state space models.

We believe that the proposed DL based pipeline can be also employed in further application areas such as grading handwritten homework of students which has also the potential to lessen the burden of lectures in grading them. On the other hand, students can also use the DL based pipeline in a similar manner in order to visualize HFCAs in their lecture notes.

ACKNOWLEDGMENT

The authors would like thank the five lecturers of the Control and Automation Engineering Department, Istanbul Technical University for their help in constructing datasets used in this paper. The authors are also grateful for the anonymous

reviewers as they provided valuable remarks and comments which have significantly enhanced the quality of a paper.

REFERENCES

- [1] D. P. Prendergast and A. M. Eydgahi, "EDCON: An educational control system analysis and design program," *IEEE Trans. Educ.*, vol. 36, no. 1, pp. 42–44, Feb. 1993, doi: [10.1109/13.204814](https://doi.org/10.1109/13.204814).
- [2] N. A. Kheir, K. J. Astrom, D. Austander, K. C. Cheok, G. F. Franklin, M. Masten, and M. Rabins, "Control systems engineering education," *Automatica*, vol. 32, no. 2, pp. 147–166, 1996, doi: [10.1016/0005-1098\(96\)85546-4](https://doi.org/10.1016/0005-1098(96)85546-4).
- [3] M. Johansson, M. Gafvert, and K. J. Astrom, "Interactive tools for education in automatic control," *IEEE Control Syst. Mag.*, vol. 18, no. 3, pp. 33–40, Jun. 1998, doi: [10.1109/37.687617](https://doi.org/10.1109/37.687617).
- [4] V. Kroumov, K. Shibayama, and A. Inoue, "Interactive learning tools for enhancing the education in control systems," in *Proc. 33rd Annu. Frontiers Educ. (FIE)*, vol. 1, Nov. 2003, pp. 23–28, doi: [10.1109/FIE.2003.1263384](https://doi.org/10.1109/FIE.2003.1263384).
- [5] A. A. Rodriguez, M. F. DeHerrera, and R. P. Metzger, "An interactive MATLAB-based tool for teaching classical systems and controls," in *Proc. Technol.-Based Re-Eng. Eng. Educ. Frontiers Educ. FIE 26th Annu. Conf.*, Salt Lake City, UT, USA, vol. 2, Nov. 1996, pp. 624–627, doi: [10.1109/FIE.1996.573031](https://doi.org/10.1109/FIE.1996.573031).
- [6] J. H. Chow and K. W. Cheung, "A toolbox for power system dynamics and control engineering education and research," *IEEE Trans. Power Syst.*, vol. 7, no. 4, pp. 1559–1564, Nov. 1992, doi: [10.1109/59.207380](https://doi.org/10.1109/59.207380).
- [7] S. Dormido, S. Dormido-Canto, R. Dormido, J. Sanchez, and N. Duro, "The role of interactivity in control learning," *Int. J. Eng. Educ.*, vol. 21, no. 6, pp. 1122–1133, Dec. 2005.
- [8] H. Vargas, J. Sanchez Moreno, C. A. Jara, F. A. Candelas, F. Torres, and S. Dormido, "A network of automatic control Web-based laboratories," *IEEE Trans. Learn. Technol.*, vol. 4, no. 3, pp. 197–208, Jul. 2011, doi: [10.1109/TLT.2010.35](https://doi.org/10.1109/TLT.2010.35).
- [9] A. Maiti, D. G. Zutin, H.-D. Wuttke, K. Henke, A. D. Maxwell, and A. A. Kist, "A framework for analyzing and evaluating architectures and control strategies in distributed remote laboratories," *IEEE Trans. Learn. Technol.*, vol. 11, no. 4, pp. 441–455, Oct. 2018, doi: [10.1109/TLT.2017.2787758](https://doi.org/10.1109/TLT.2017.2787758).
- [10] S. D. Bencomo, "Control learning: Present and future," *Annu. Rev. Control*, vol. 28, no. 1, pp. 115–136, Jan. 2004, doi: [10.1016/j.arcontrol.2003.12.002](https://doi.org/10.1016/j.arcontrol.2003.12.002).
- [11] J. Sanchez, S. Dormido, R. Pastor, and F. Morilla, "A Java/MATLAB-based environment for remote control system laboratories: Illustrated with an inverted pendulum," *IEEE Trans. Educ.*, vol. 47, no. 3, pp. 321–329, Aug. 2004, doi: [10.1109/TE.2004.825525](https://doi.org/10.1109/TE.2004.825525).
- [12] A. Leva and F. Donida, "Multifunctional remote laboratory for education in automatic control: The CrAutoLab experience," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2376–2385, Jun. 2008, doi: [10.1109/TIE.2008.922590](https://doi.org/10.1109/TIE.2008.922590).
- [13] R. C. Dorf and R. H. Bishop, *Modern Control Systems*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2011.
- [14] *MATLAB Control System Toolbox Users Guide*, MathWorks, Inc., Natick, MA, USA, 2019.
- [15] X. Ding, "Chinese character recognition," in *Handbook of Pattern Recognition & Computer Vision*, vol. 1, C. H. Chen, Ed., 3rd ed. Singapore: World Scientific, 2005, ch. 3.2, pp. 241–257.
- [16] L.-N. Teow and K.-F. Loe, "Robust vision-based features and classification schemes for off-line handwritten digit recognition," *Pattern Recognit.*, vol. 35, no. 11, pp. 2355–2364, Nov. 2002, doi: [10.1016/S0031-3203\(01\)00228-X](https://doi.org/10.1016/S0031-3203(01)00228-X).
- [17] L. B. Kara and T. F. Stahovich, "An image-based, trainable symbol recognizer for hand-drawn sketches," *Comput. Graph.*, vol. 29, no. 4, pp. 501–517, Aug. 2005, doi: [10.1016/j.cag.2005.05.004](https://doi.org/10.1016/j.cag.2005.05.004).
- [18] M. Bresler, T. V. Phan, D. Prusa, M. Nakagawa, and V. Hlavac, "Recognition system for on-line sketched diagrams," in *Proc. 14th Int. Conf. Frontiers Handwriting Recognit.*, Heraklion, Crete, Sep. 2014, pp. 563–568.
- [19] Q. Chen, D. Shi, G. Feng, X. Zhao, and B. Luo, "On-line handwritten flowchart recognition based on logical structure and graph grammar," in *Proc. 5th Int. Conf. Inf. Sci. Technol. (ICIST)*, Changsha, China, Apr. 2015, pp. 424–429.
- [20] T. Hammond and B. Paulson, "Recognizing sketched multistroke primitives," *ACM Trans. Interact. Intell. Syst.*, vol. 1, no. 1, pp. 4:1–4:34, Oct. 2011, doi: [10.1145/2030365.2030369](https://doi.org/10.1145/2030365.2030369).
- [21] L. B. Kara and T. F. Stahovich, "Hierarchical parsing and recognition of hand-sketched diagrams," in *Proc. ACM SIGGRAPH Courses (SIGGRAPH)*, Santa Fe, NM, USA, 2004, pp. 13–22.
- [22] R. Mörzinger, R. Schuster, A. Horti, and G. Thallinger, "Visual structure analysis of flow charts in patent images," in *Proc. CLEF (Online Work. Notes/Labs/Workshop)*, 2012, pp. 629–633.
- [23] B. G. Vasudevan, S. Dhanapanichkul, and R. Balakrishnan, "Flowchart knowledge extraction on image processing," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intell.)*, Hong Kong, Jun. 2008, pp. 4075–4082.
- [24] Y. Yu, A. Samal, and S. C. Seth, "A system for recognizing a large class of engineering drawings," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 8, pp. 868–890, Aug. 1997.
- [25] M. S. Iqbal, B. Luo, R. Mehmood, M. A. Alrige, and R. Alharbey, "Mitochondrial organelle movement classification (fission and fusion) via convolutional neural network approach," *IEEE Access*, vol. 7, pp. 86570–86577, 2019, doi: [10.1109/ACCESS.2019.2925041](https://doi.org/10.1109/ACCESS.2019.2925041).
- [26] Q. Lu, C. Liu, Z. Jiang, A. Men, and B. Yang, "G-CNN: Object detection via grid convolutional neural network," in *IEEE Access*, vol. 5, pp. 24023–24031, 2017, doi: [10.1109/ACCESS.2017.2770178](https://doi.org/10.1109/ACCESS.2017.2770178).
- [27] A. Beke and T. Kumbasar, "Learning with type-2 fuzzy activation functions to improve the performance of deep neural networks," *Eng. Appl. Artif. Intell.*, vol. 85, pp. 372–384, Oct. 2019, doi: [10.1016/j.engappai.2019.06.016](https://doi.org/10.1016/j.engappai.2019.06.016).
- [28] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *Proc. 7th Int. Conf. Document Anal. Recognit.*, Aug. 2003, pp. 958–963.
- [29] X.-Y. Zhang, Y. Bengio, and C.-L. Liu, "Online and offline handwritten Chinese character recognition: A comprehensive study and new benchmark," *Pattern Recognit.*, vol. 61, pp. 348–360, Jan. 2017, doi: [10.1016/j.patcog.2016.08.005](https://doi.org/10.1016/j.patcog.2016.08.005).
- [30] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: [10.1109/5.726791](https://doi.org/10.1109/5.726791).
- [31] H. Ling, J. Wu, L. Wu, J. Huang, J. Chen, and P. Li, "Self residual attention network for deep face recognition," *IEEE Access*, vol. 7, pp. 55159–55168, 2019, doi: [10.1109/ACCESS.2019.2913205](https://doi.org/10.1109/ACCESS.2019.2913205).
- [32] Y. Wang, B. Lei, A. Elazab, E.-L. Tan, W. Wang, F. Huang, X. Gong, and T. Wang, "Breast cancer image classification via multi-network features and dual-network orthogonal low-rank learning," *IEEE Access*, vol. 8, pp. 27779–27792, 2020, doi: [10.1109/ACCESS.2020.2964276](https://doi.org/10.1109/ACCESS.2020.2964276).
- [33] X. Wang, X. Wang, and S. Mao, "Deep convolutional neural networks for indoor localization with CSI images," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 316–327, Jan./Mar. 2020.
- [34] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019, doi: [10.1109/ACCESS.2019.2912200](https://doi.org/10.1109/ACCESS.2019.2912200).
- [35] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [37] S. Jialin Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.



DORUKHAN ERDEM received the B.Sc. degree in control and automation engineering from Istanbul Technical University, in 2018, where he is currently pursuing the M.Sc. degree with Control and Automation Engineering Department, Faculty of Electrical and Electronics Engineering. He is working as a Research Assistant with Control and Automation Engineering Department, Faculty of Electrical and Electronics Engineering, Istanbul Technical University. His research interests include computational intelligence, machine learning, deep learning, and system modelling in the area of applied robotics.



AYKUT BEKE (Student Member, IEEE) received the B.Sc. and M.Sc. degrees in control and automation engineering from Istanbul Technical University, in 2015, where he is currently pursuing the Ph.D. degree with Control and Automation Engineering Department, Faculty of Electrical and Electronics Engineering. He is working as a Research Assistant with the Control and Automation Engineering Department, Faculty of Electrical and Electronics Engineering, Istanbul Technical

University. His research interests include control theory, cognitive robotics, computational intelligence, machine learning, and time series forecasting with special interest in fuzzy logic theory.



TUFAN KUMBASAR (Senior Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in control and automation engineering from Istanbul Technical University.

He is currently an Associate Professor with Control and Automation Engineering Department, Faculty of Electrical and Electronics Engineering, Istanbul Technical University. He has currently authored more than 100 articles in international conferences, journals, and books. His major research interests are in computational intelligence, notably type-2 fuzzy logic, fuzzy control, neural networks, evolutionary algorithms, and control theory. He is also interested in process control, robotics, intelligent control, and their real-world applications.

Dr. Kumbasar received the Best Paper Awards from the IEEE International Conference on Fuzzy Systems, in 2015, and the Sixth International Conference on Control Engineering and Information Technology, in 2018. He has served as a Publication Co-Chair, the Panel Session Co-Chair, the Special Session Co-Chair, a PC, an IPC, and a TPC in various international and national conferences. He is an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS and an Area Editor of the *International Journal of Approximate Reasoning*.

• • •