An Application of the Hough Transform and Convolutional Neural Networks to Detect Straight Lines

Moussa Bamogo[1(✉)] and Abdoulaye Sere[2]

[1] ER-SIC, LAMDI, Universite Nazi BONI, Bobo Dioulasso, Burkina Faso
bmgm25@gmail.com

[2] Reseau des Enseignants Chercheurs et Chercheurs en Informatique du Faso (RECIF), Ouagadougou, Burkina Faso
abdoulaye.sere@recifaso.org
https://www.recifaso.org

Abstract. The topic addressed in this research study concerns the combination of the Hough Transform with convolutional neural networks to improve pattern recognition in a collection of images. We propose a neural network model that takes as input a collection of images. In order to be able to compare the results obtained, the collection is processed by the Extended Hough Transform on the one hand, and on the other hand, it has not undergone any processing by the Extended Standard Hough Transform. The model proposed in our approach is composed of a set of convolutional layers and a fully connected layer. For this study, we used a dataset containing a total of 10,200 images. The experimental results obtained with our model give an accuracy of 70.00% with the dataset treated with the Extended Hough Transform and 66.67% with the other dataset. It can also detect images containing lines. In view of the experiments carried out, we have seen that the size of the learning base and the material resources are key factors in obtaining better results. Hough Transforms help to improve the accuracy of the convolutional neural network.

Keywords: Convolutional Neural Networks · Hough Transform · Pattern Recognition

1 Introduction

The Hough Transform is an image processing technique that was initially developed to detect geometric shapes in an image, such as straight lines or circles. Over the years, it has been adapted to solve various computer vision problems. The Hough Transform enables the detection of lines and curves in an image by converting points in the image to a parametric representation in a voting space.

Nowadays, with the advent of big data, processing large volumes of data is essential. They are found in many fields, such as medical imaging, cartography,

K. Cheikh M. F. Kebe et al. (Eds.): InterSol 2024, LNICST 610, pp. 115–125, 2025.
https://doi.org/10.1007/978-3-031-86493-3_10

biomedical research, research laboratories, computer vision, and systems monitoring [26]. Serval research has been done on image processing, and different techniques have been implemented. Among these, we can cite Hough Transform and deep learning, which make it possible to recognize a shape in an image. Many efforts have been agreed upon in the literature with a view to optimize them. But building an automatic system that recognizes shapes in an image remains a challenge.

The problem addressed in this study is how to improve pattern recognition in image collections by combining the Hough Transform with convolutional neural networks (CNN). In other words, the study focuses on how to exploit the advantages of classical image processing techniques such as the Hough Transform by effectively integrating them with modern deep learning approaches like CNNs to obtain better results in terms of pattern recognition accuracy.

The central question we want to solve is, How can integrating the Hough Transform with convolutional neural networks improve the accuracy of pattern recognition in a collection of images?

The hypothesis we pose in this study is that Integration of the Hough Transform with convolutional neural networks will improve the accuracy of pattern recognition in a collection of images compared to the exclusive use of convolutional neural networks without this integration.

Learning is a field of artificial intelligence that attempts to mimic the neural functioning of the human brain through an artificial neural network. It appeared in 1980, following the work of Yann Lecun [1]. Together with two other computer scientists, they developed a particular type of algorithm called a convolutional neural network. The use of this algorithm has become widespread with the emergence of a new generation of graphics chips (GPU) capable of performing more than a trillion operations per second and with the availability of large amounts of computer data. Deep learning has caused a major upheaval in the field of artificial intelligence and its applications. He has also contributed to many applications, such as image recognition and classification [2] and automatic processing of natural languages [2]. This progress is explained by the availability of large amounts of learning data produced by sensors, surveys, social networks, and the Internet of Things in various types and formats. This massive amount of data of various types and formats is known as big data. This data allows artificial intelligence to understand and learn like humans think.

Convolutional neural networks (CNN) are a type of acyclic artificial neural network that can automatically extract features from input images while remaining invariant to slight image distortions. They also implement the notion of weight sharing, making it possible to considerably reduce the number of network parameters. This makes it possible to considerably count the local correlations contained in an image. These networks were inspired by the work of [7,8], and [9]. The weights are forced to be equal to detect lines, points, or corners at all possible locations in the image, effectively implementing the idea of weight sharing [10]. In these works [11], CNNs are used for handwritten character recognition. They are used in image and video recognition, recommendation systems, and natural language processing [11–13]. The more data a CNN receives, the more it learns and the more accurate it becomes. What about Hough's transformation? The Hough Transform, which is a technique for recognizing geometric shapes in a digital image, was introduced in the 1960s by Paul V.C. Hough [3] with the aim of detecting alignments (straight lines). This technique has been extended to the detection of other geometric shapes thanks to the work of Rosenfeld [4] and Duda et al. [5]. With [6,14–19], it was generalized to the detection of arbitrary and complex geometric shapes. The work [20–23] proposed several methods of applying pattern recognition by Hough Transform. The Extended Standard Hough Transform (ESHT) is a method introduced by SERE et al. [25] that can detect discrete straight lines that are seen as a sequence of pixels. For the reconstruction of a discrete line into a continuous line, two essential notions have been introduced: dual and preimage. Dual is an equivalent of the Hough Standard Transformation in ESHT. Algorithm 1 describes the sequence of discrete, naive, and standard straight-line recognition by the Extended Standard Hough Transform (ESHT).

Algorithm 1. Naive and standard discrete straight-line recognition

data : A set S of n pixels $P_1; P_2; ...; P_n$
Begin
Preimage ← Dual(P_1)
$i \leftarrow 2$
while Preimage $\neq \emptyset et i \leq$n **do**
  Preimage ← Preimage ∩ Dual(P_i)
  $i \leftarrow i + 1$
end while
if Preimage $\neq \emptyset$ **then**
  S belongs to a line
else
  S does not belong to a line
end if
End

In this paper, we are interested in pattern recognition in an image database using deep learning. This is for us to propose an algorithmic approach to processing a collection of images while combining the Hough Transform and convolutional neural networks. This document is organized as follows: in Sect. 2, method description; in Sect. 3, the experimental results; and in Sect. 4, the conclusion and perspectives.

2 Method Description

The objective of our study is to increase the prediction rate of a convolutional neural network while reducing the failure rate in a collection of images. To achieve this objective, we will use ESHT to preprocess the images and then pass them through a predictive model based on convolutional neural networks. In this section, we present our approach in the context of this study and the test data. Our model is made up of three convolution layers, a Max Pooling layer placed after each convolutional layer, and a fully connected layer. The model receives a collection of images as input. The images are fed into the first convolution layer, which is composed of filters of size 3×3, with a ReLU correction function applied to each layer. A matrix called a feature map is created at the output of this layer. We then pass this matrix through the pooling layer (Max Pooling) in order to reduce the size of this matrix and only keep the important information. The matrix obtained at the output is given as input to the second convolution layer, which has the same characteristics as the first. Indeed, it is composed of filters and implements a ReLU correction function. This is followed by a max-pooling layer. The same processing is followed as in the previous one. The third convolution layer is composed of filters of size 3×3 and implements a ReLU correction function. The values of the last obtained feature map are passed to the last Max Pooling layer. The values obtained by this last Max Pooling layer are concatenated into a single vector and transmitted to a fully connected layer. The input vector values are transformed to return a new output vector. This last vector contains as many elements as there are classes: each element of the vector indicates the probability for the image to belong to a class.

These probabilities are calculated by the fully connected layer, which uses a sigmoid function, in our case, as the activation function. A loss function, or cost function, is associated with the fully connected layer to measure the error between the network prediction and the actual data annotation. For updating the model in order to minimize the cost function and have better predictions, we chose Adam, which is an optimization algorithm. This choice is justified by a comparative study of the simulations extracted in [24]. Bringing together different methods shows that for the same optimization problem and the same learning rate, adam tends to make the error converge towards zero much more quickly. The Fig. 1, Algorithm 2 and Algorithm 3 illustrate our method.

Algorithm 2. How our method works

données : dataset a collection of images $I = \{I_1; I_2; ...; P_N\}$
Begin
Function $modelCNN(dataset)$:
for all $I_k \in dataset$ **do**
  ConvolutionRelu()
  MaxPooling()
  ConvolutionRelu()
  MaxPooling()
  ConvolutionRelu()
  MaxPooling()
  ConnectedLayer()
end for
Return Prediction()
End

Algorithm 2 describes the process of our CNN model, which takes a set of images as input and then carries out processing return predictions.

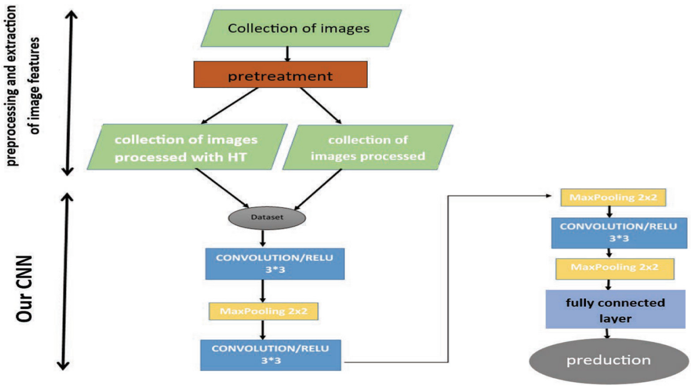


Fig. 1. Illustration of our method Hough Transform and CNN

Algorithm 3. Process of data preparation and processing with EHST

---

data : A collection of images $I = \{I_1; I_2; ...; P_N\}$
f : A file containing images
Begin
Function $collect(Collection \quad I)$:Function saving images to a file
for all $I_k \in I$ **do**
  if I_k **then**
    CreateEmpty file f
    OpenFile(f) "in writing"
    AddInFile(f, I_k)
    CloseFile(f)
  end if
end for
Function $traitemenTH(f)$:Processing function with EHST
OpenFile(f) "read"
for all $I_k \in f$ **do**
  if I_k **then**
    FunctionEHST(I_k)
    CreateEmpty file $f2$
    OpenFile($f2$) "in writing"
    AddInFile($f2, I_k$)
    CloseFile($f2$)
  end if
end for
End

Algorithm 3 illustrates data preparations with the Extended Standard Hough Transform.

3 Experimental Results

3.1 Working Environment

Hardware and Operating System. An HP computer with the following specifications has been used to carry out the simulations:

– Processor: Intel(R) Core(TM) i5 CPU 2.56 GHz. Item RAM: 6 Go
– Operating system: Windows 10, 64 bits.

Programming Tools and Language. Python has been selected as the simulation's chosen software environment. For programming, graphical visualization, and numerical computation, Python is an effective tool. Being object-oriented and having dynamic semantics, it is a high-level interpreted programming language, meaning there is no compilation step. Developers and programmers in general have embraced Python worldwide. Code maintenance costs can be minimized due to the language's ease of learning and simplicity.

Open Source Computer Vision (Open CV) is a free library. Originally developed by Intel, specializing in real-time image processingreal. We used this tool in the data preprocessing phase. Shewas used to more precisely implement the techniques of image normalization, zooming, rotation, and translation. We used Keras for its power and efficiency. It is an open-source library that allows rapid experimentation with neural network models and is very simple to use. It is a Python library that encapsulates access to functions offered by several machine learning libraries, including Tensorflow. It was used to implement our learning model.

3.2 Dataset

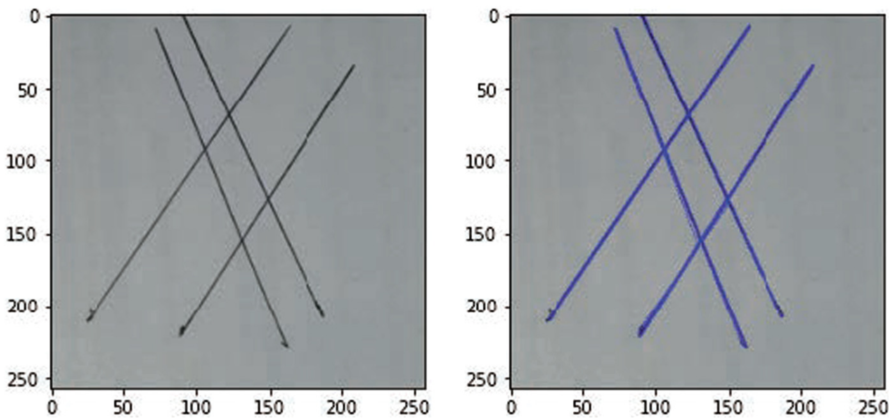
We carried out our experiments on a data set that we created. This database contains images of size 32×32. These images belong to two classes, namely those containing straight lines and those not containing any. We drew these images ourselves. Data augmentation is a method used in deep learning to enlarge the size of data when the dataset is not consistent. This operation duplicates the data by applying certain geometric transformation operations in image processing in order to prevent the same data from being repeated in the data set. Since our dataset does not contain enough images, this operation is very important. In our context, it is based on image processing applications. The data processing consisted, first of all, of duplicating our dataset to have two batches. Then we proceeded to process the second batch with the Extended Hough Standard Transform. This made it possible to improve the images by clearly marking the presence of the straight lines. Figure 2 shows an image processed with the

Extended Hough Standard Transform (EHST) and another unprocessed. The organization of the data consisted of making a mixture of all the labels collected, namely the images containing straight lines and the images not containing any. The goal is to slice, then split, and define the data for training, testing, and validating the neural network model. Indeed, for training our model, we used a total of two hundred (100) simulated images in our dataset. Then, we applied deep learning data augmentation methods mentioned above to reach 10,200 images. After the augmentation technique, we divided this set into three subsets, including 70% of the data for training, 15% for testing, and 15% for model validation.

3.3 Evaluation Metrics

Metrics allow define indicators to evaluate the performance of a learning model. These are the confusion matrix, precision, recall and f-measure. To find out the type of errors made, we use the confusion matrix which is a summary of the prediction results on a classification problem. Correct and incorrect predictions are divided by class (Table 1):

Precision is the proportion of relevant elements among all the elements proposed. It is calculated using the formula $P = \frac{VP}{VP+FP}$ where VP is the number of true positives and FP that of false positives. The recall which represents the proportion of relevant elements proposed among all the relevant elements is cal-



(a) image (b) image processed with ESHT

Fig. 2. Illustration of processing with Hough Transformation

Table 1. Confusion matrix

		Positive	Negative
Predicted Class	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

culated with the formula $R = \frac{VP}{VP+FN}$, FN being the number of false negatives. The F-measure is a measure that combines precision and recall. This is their average. The formula is as follows

$$F - mesure = 2 * \frac{(Precision * Recall)}{(Precision + Recall)}$$

3.4 Results

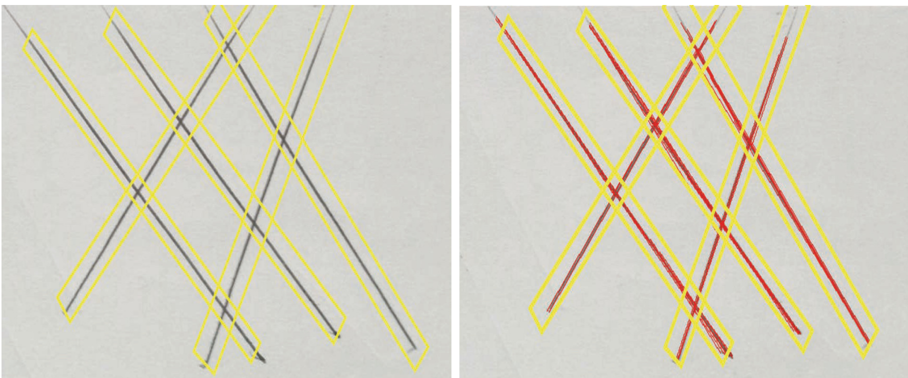
The Table 2 below summarizes the overall results obtained with our model. There is an improvement in accuracy with the dataset that has been processed with Hough Transformation.

Figures 3 and 4 show the results of detecting lines with our method. The Fig. 3 shows the results of detecting lines.

Figure 4 shows a comparison of straight line detection with our method. The first Fig. 4a of the material illustrates the detection of lines without Hough Transform. Figure 4b shows the visual output with our method, we see more vertical and horizontal lines clearly detected.

Table 2. Summary table of the different results

	Without HT	with HT
Prediction	66.67%	70.00%
Loss	33,33 %	30,00 %



(a) without hough transform (b) with hough transform and CNN

Fig. 3. Illustration of detection with hough transform and CNN

(a) without hough transform (b) with hough transform and CNN

Fig. 4. Illustration of detection with hough transform and CNN

3.5 Discussions

The results obtained after evaluation of classification model give an accuracy rate of 66.67% and 33.33% without processing with Extended Hough Standard Transform, compared to 70.00% and 30.00% with the dataset processed with Extended Hough Standard Transform. After an analysis of the confusion matrices, we found that the classifier of our model recognizes the images containing straight lines in the test data of the two batches of the dataset. Using the Hough Transform improves detection accuracy. The first difficulty lies in the availability of real data. We simulated and collected the images by taking photos with our smartphones, which undoubtedly affects the quality of the images. Added to this is the memory space capacity that we have. The difficulty is the low capacity of the memory space in our work environment. This is why we used images in 32×32 format. In fact, training a CNN model requires larger memory space.

4 Conclusion and Perspectives

In this paper, we propose a methodology for the recognition of shapes in an image by coupling the Hough transform and deep learning. A state-of-the art of previous works and methods on the Hough Transform and deep learning are reported. This guided us to make a goodchoice of a methodology adapted to the problem, as well as for the design of the recognition model. We have proposed a convolutional neural network that takes as input image data and preprocessed image data with theExtended Standard Hough Transform. These different technologies have a great ability to solve complex problems in the fields of computer vision and pattern recognition. However, their userequires a large amount of data and more powerful computing machines to perform and produce better results. In our model, the convolutional part was used to extract features from the images, and the fully connected layer part was used for prediction. For the evaluation of our approach, we used images that we have collected, and the experimental results obtained with the model have been illustrated. In fact, the accuracy of our model was 70.00% with the dataset that was processed with the Extended Standard Hough Transform and 66.67% with the other dataset that was not processed treated with the Extended Standard Hough Transform. The model is used to classify images containing lines. In perspective, we will extend our approach by diversifying the data sources, strengthening the model, proposing an application in transport.

References

1. LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., Jackel, L.: Handwritten digit recognition with a back propagation network. In: Advances in Neural Information Processing Systems 2, pp. 396–404, 219. Morgan Kaufmann, San Francisco (1990)
2. Zbontar, J., LeCun, Y.: Stereo matching by training a convolutional neural network to compare image patches. Mach. Learn. Res. 17–49 (2016)
3. Hough, P.V.C.: Method and means for recognazing complex patterns, US3283070A (1960)
4. Rosenfeld, A.: Progress in picture processing: 1969–71. ACM Comput. Surv. **5**, 81–108 (1973)
5. Duda, R.O., Hart, P.E.: Use of the hough transformation to detect lines and curves in pictures, W. Newman, University of Nevada (1972)
6. Ballard, D.H.: Generalizing the hough transform to detect arbitrary shapes. Pattern Recognit. **13**, 111–122 (1981)
7. Hubel, D.H., Wiesel, T.N.: Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. J. Physiol. **160**, 106–154 (1962)
8. Fukushima, K.: Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybern. **36**, 193–202 (1980)
9. Fukushima, K.: A neural network model for selective attention in visual pattern recognition. Biol. Cybern. **55**, 5–15 (1986)
10. Rumelhart, D.E., McClelland, J.L.: Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations. The MIT Press (1986)
11. Denke, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D., LeCun, Y., Boser, B.: Handwritten Digit Recognition with a Back-Propagation Network. BibSonomy. https://www.bibsonomy.org/bibtex/d0f161d61285aca3b30c3add9416921e
12. Collobert, R., Weston, J.: A unified architecture for natural language processing: deep neural networks with multitask learning, pp. 160–167. ACM Press (2008)
13. Sutskever, I., Hinton, G.E., Krizhevsky, A.: ImageNet Classification with Deep Convolutional Neural Networks, pp. 1097–1105. Curran Associates, Inc. (2012)
14. Maitre, H.: Un panorama de la transformation de Hough. Traitement du Signal **2** (1985)
15. Duda, R.O., Hart, P.E.: Use of the hough transformation to detect lines and curves in pictures. W. Newman (1972)
16. Merlin, P.M., Farber, D.J.: A parallel mechanism for detecting curves in pictures. IEEE Trans. Comput. **C-24**, 96–98 (1975)

17. Shapiro, S.D.: Feature space transforms for curve detection. Pattern Recognit. **10**, 129–143 (1978)
18. Shapiro, S.D.: Transform method of curve detection for textured image data. IEEE Trans. Comput. **C-27**, 254–255 (1978)
19. Shapiro, S.D.: Transformations for the computer detection of curves in noisy pictures. Comput. Graph. Image Process. **4**, 328–338 (1975)
20. Kälviäinen, H., Hirvonen, P., Xu, L., Oja, E.: Comparisons of probabilistic and non-probabilistic hough transforms, vol. 801, pp. 350–360. Springer, Heidelberg (1994)
21. Sere, A., Coulibali, L., Diarra, M., Sie, O., Ouedraogo, F.T.: An Application of the Triangular Hough Transform and the Rectangular Hough Transform in Noisy Analytical Straight Line Recognition. Africomm 2015 (2015)
22. Sere, A., Sie, O., Traore, S.: Extensions of standard hough transform based on object dual and application. J. Emerg. Trends Comput. Inf. Sci. **6** (2015)
23. Sere, A., Sie, O., Andres, E.: Extended standard hough transform for analytical line recognition. Int. J. Adv. Comput. Sci. Appl. **4** (2013)
24. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Conference paper at the 3rd International Conference for Learning Representations, San Diego (2014)
25. Sere, A., Sie, O., Andres, E.: Extended standard hough transform for analytical line recognition. In: 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT). IEEE Xplore (2012)
26. Dash, S., Shakyawar, S.K., Sharma, M., Kaushik, S.: Big data in healthcare: management, analysis and future prospects. J. Big Data **6** (2019)