STM JOURNALS

International Journal of
Radio Frequency Innovations

Volume 2, Issue 1, 2024
January–June
DOI (Journal): 10.37591/IJRFI

https://journals.stmjournals.com/ijrfi

Review IJRFI

Advancements in Handwriting Recognition: A Deep Learning Approach

Sreya Adhikary[1,] *, Arnab Das[2], Sankhadip Chowdhury[3]

Abstract

This article provides detailed information about handwriting text recognition. Some human characteristics are unique to the individual. Writing is one of the scientifically proven habits that is different for everyone. Handwriting Text Recognition (HTR) is responsible for identifying written characters and converting them into digital text. HTR is an intensively researched area, but improvements can still be made in accuracy and efficiency. Digitization of manuscripts is very useful in today's world because it allows information to be easily accessed anytime and anywhere. Digitized books can be used for commercial purposes and are safer and more environmentally friendly than textbooks. This review will highlight the various applications made so far in this field along with their advantages, limitations, and realities. Considering the amount of data collected in the human industry, optical character recognition (OCR) of data is very useful. Optical recognition is a scientific method that can transform various types of data or images into identifiable, editable and searchable data. Over the years, researchers have used AI/machine learning tools to automatically analyze written and printed documents to convert them into electronic documents. The purpose of this article is to conduct an extensive investigation of the various deep convolutional neural networks that can provide improved accuracy for offline cursive handwriting recognition for English language. The research methodology with the following five different phases such as acquiring the data, pre-processing, feature extraction, classification and finally the evaluation and model enhancement was applied for this study.

Keywords: OCR, HTR, Deep Learning, CNN, Digitization

INTRODUCTION

A vital technological advancement, handwriting recognition can be used for everything from digitising old texts to promoting human-computer connection. Conventional techniques depended on manually created characteristics and statistical models, which frequently found difficulty accounting for the intricacy and variety of handwritten text [1]. Deep learning has completely changed this field by providing strong tools to automatically extract characteristics from unprocessed data and increase recognition accuracy. This article explores the deep learning-driven advances in handwriting recognition, highlighting the important architectures, techniques, and datasets that have influenced the state-of-the-art.

***Author for Correspondence**
Sreya Adhikary
E-mail: sreyaadhikary316@gmail.com

[1]Student, Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), Institute of Engineering and Management, Kolkata, West Bengal, India.

Received Date: May 31, 2024
Accepted Date: June 27, 2024
Published Date: July 01, 2024

Citation: Sreya Adhikary, Arnab Das, Sankhadip Chowdhury. Advancements in Handwriting Recognition: A Deep Learning Approach. International Journal of Radio Frequency Innovations. 2024; 2(1): 28–34p.

Deep Learning Frameworks for Recognising Handwriting

CNNs, or convolutional neural networks: CNNs' capacity to recognise spatial hierarchies in images has led to their widespread adoption in handwriting recognition applications. Yann LeCun's architectures, including LeNet, were some of the first to show how well CNNs could recognise handwritten numbers.

Recurrent Neural Networks (RNNs): RNNs are useful for handwriting recognition applications

where stroke sequencing is important since they can handle sequential data well, especially Long Short-Term Memory (LSTM) networks. Strong models like the CRNN (Convolutional Recurrent Neural Network), which make use of both spatial and temporal information, are the result of combining CNNs with RNNs.

Attention Mechanisms: By enabling models to concentrate on pertinent segments of the input sequence, attention mechanisms have significantly improved handwriting recognition. Transformer-based models have demonstrated promising performance in handwriting recognition tasks, especially in languages with intricate scripts. These models rely on self-attention [2–5]. Training datasets can be enhanced by using Generative Adversarial Networks (GANs), which have been investigated for their potential to produce synthetic handwriting data. This is especially helpful in situations where there is a dearth of labelled data [6].

Background

Literacy is a powerful tool that can be used in many ways, from analyzing data to recognizing letters and text. In this project, we developed a custom TensorFlow model to extract text from hand-written images using the IAM dataset. Our design model can be interpreted with poor performance against the competition using CNN. Along the way, we discuss various ways to improve model performance, such as fine-tuning hyperparameters, using different data or support, evaluating different models, or combining additional features. Most organizations use data to obtain information from customers. This information is usually written by hand. This information can be stored on paper, drawings, etc. To store or collect information more easily, information is converted and stored in digital format. One way to process this information is to manually populate the computer with the same information. Manually processing data can be frustrating and time- consuming. Therefore, a special text recognition software is needed to recognize the texts in image files [9,10]. Handwriting recognition software makes it easy to retrieve information from written documents and store them in electronic format. Banking, healthcare, and many other organizations use this information frequently.

Problem Statement

Although there are many writing tools, many people still prefer to write traditionally, using paper and pen. But writing has its drawbacks. Physical data is difficult to store and access effectively, search for, and share with others effectively. As a result, much important information is lost or overlooked because the information is not converted into digital format. That's why we thought to solve this issue in our project because we decided that creating digital information is easier than documents; This will help people better access, analyze and search their data while still using the information they have consented to. Literacy is the phenomenon of transforming written words into readable form. One of the biggest problems with writing skills is the differences in writing style, it can be totally different for different authors. The purpose of typing is to use a computer-assisted user who can manipulate symbols in written materials and digitize and translate written words for machine reading. Written characters are divided into Internet characters. This is a system that identifies texts as they are created.

Aims and Objectives

The purpose of this project is to further investigate the work of classifying manuscripts and converting manuscripts into digital forms. Writing is a broad spectrum, and we hope to narrow the scope of our project by showing what writing means. The challenge we faced in this project was to segment the image of the text, which could be text or block text. This project can be mixed with an algorithm that splits a text image into a single image, which can be combined with an algorithm that splits a single image into a full image. Thanks to these layers, our project can deliver the document tothe end user and become an effective model that helps users solve the problem of digital formatting of data collection, modification by taking photographs of written pages. Based on findings using all images, we aim to improve by extracting characters from every word image and then recreating the whole word by splitting each character on its own. So, in a few words, our model takes a picture of a word and extracts the name of the word. Even though some layers needed to be added atop our model to bring full functionality to end users, we think that the most impressive and challenging part of this problem is distribution, so we

decided to solve this problem. To solve the issue, we use the full image for every word since CNNs are likely to work better on raw pixels in place of characteristics or parts of the image.

METHODOLOGY

Acquisition of Information: The first stage of the process involves obtaining the necessary information; in this case this involves collecting image data from the IAM repository. The library will contain a collection of images relevant to the specific task or project at hand. After this, the main step in the data preparation process is to split the captured images into different parts. First, this requires splitting the image dataset into two large groups: training and testing. The purpose of this segmentation is to create separate files that serve different functions in the pipeline. The training method forms a large part of the material used to train machine learning models. By providing this training data, the model can learn patterns, features, or features present in images, thereby improving its ability to make accurate or logical decisions at first guess when faced with new, invisible information. The testing process, on the other hand, is a way to evaluate the effectiveness and potential of the educational model [7,8]. This method contains images that are not part of the training set and therefore the model cannot identify them from the previous data. By evaluating the model's performance on these independent test data, researchers can evaluate the model's performance in predicting or classifying real-world data [9].

Data pre-processing: Pre-processing is an important stage in the data preparation process and is important to ensure the quality and reliability of the data set. As shown in previous research, this planning phase should be used to correct inconsistencies and model the material. By taking this first step, the goal is to reduce classification issues that can affect recognition while improving correct behavior [10]. The main purpose of preprocessing is to eliminate noise, including removing or reducing unnecessary or irregular objects in the image data. These artifacts often appear as flaws or disturbances that can affect the performance of the underlying machine learning algorithm. By using noise removal techniques such as filtering or smoothing algorithms, irrelevant points in the data can be removed, thus increasing the efficiency of analysis and interpretation.

Another important part of preprocessing is tilt correction, which takes into account distortions in the orientation or tilt of characters in the image. This change may occur for many reasons, including the discovery of flaws or inaccuracies in the manuscript [11,12]. By correcting skew distortion, characters can be optimized with machine learning models, making them more recognizable and consistent. Additionally, size normalization is the first important step to measure the length of characters in the data. Slight differences may occur due to changes in model, scanning resolution, or image quality. Size normalization techniques, such as resizing or rescaling, adjust characters equally to sizes to ensure consistency and comparability across the entire dataset.

Feature extraction: Loops, curves, vertical or horizontal lines etc. A bunch of convolutions are undertaken in this step to extract characteristics and other features. The convolution is done by the kernels or filters required for the algorithm to work. Packaging is ready. Create custom menus with appropriate features. When used together, you can control your excess weight and get a good workout in a short time. Pooling also reduces the power consumption required for data processing by reducing the amount of data. Additionally, the method supports extraction of rotation- and position-independent control features. Shared images are flattened into vectors.

Classification: Length vector input to the neural network. A fully connected neural network then uses these functions and sends the votes to the class. Finally, forward and reverse training is used repeatedly to establish a neural connection with the tool and weight. Since choosing the wrong hyperparameter can lead to low interpretation and high computational expenses, we present a comprehensive analysis of the use of popular CNNs and hybrid models with appropriate hyperparameters.

Evaluation and model development: Feature extraction technology, better model training and classifiers are the foremost components that determine the accuracy of the documentation system. An extensive evaluation of the model using different hybrid architectures and CNN is presented. Various

assumptions are used to evaluate the accuracy of CNN architectures. The results were compared with various literature/studies. In addition, the best results are achieved by combining designs. To improve recognition accuracy, the following methods are provided when necessary:

- Increase the input size of the neural network so that complete text lines could be utilized
- Use decoding based on word beam search so that the output is constrained to the dictionary words
- Increase the number of convolutional neural network layers
- Deploy text correction by searching for the most similar word, in case if it is not recognized
- Apply data augmentation by increasing the size of the data set using random transformations to the input.

Proposed Architecture

The CNN architecture used in this study is shown in Figure 1. First, a weight matrix is applied to the input image and enhanced to display specific features while preserving spatial details. This convolution technique reduces the number of false positives by reducing the number of pixels in the image. Layer pooling is used to further reduce the image size without changing the depth. CNN architecture has three convolutional layers, each with appropriate stepping, padding and boosting functions to improve feature extraction. Additionally, our layers have been merged to reduce the size of the map. Two thick layers are added after the convolution and pooling layers, followed by a final output layer with SoftMax activation to facilitate segmentation.

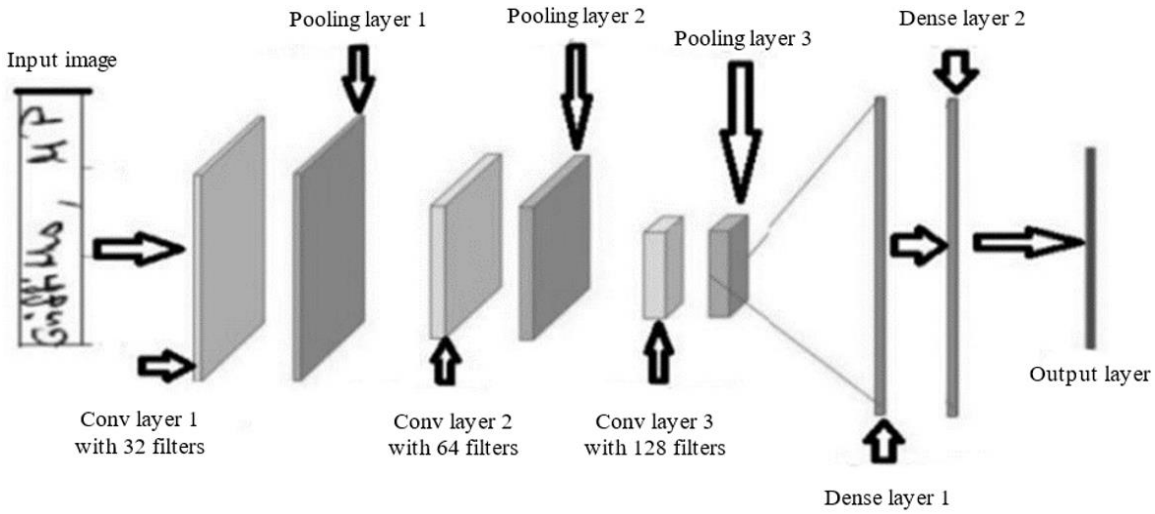

Figure 1. The Proposed CNN Architecture.

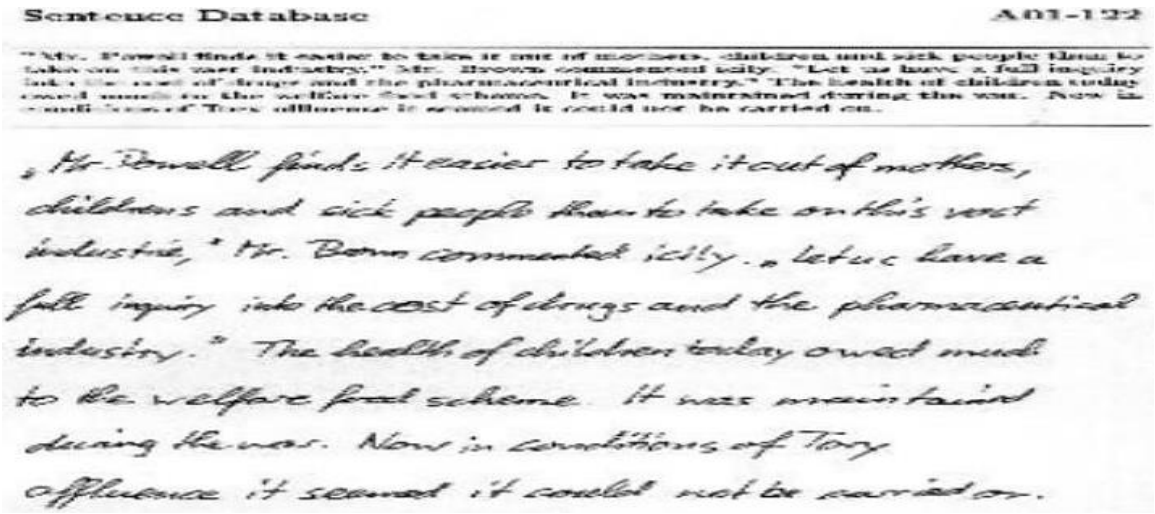

Figure 2. Image of Sentence from IAM Dataset.

Data collection and research: The project used sentences from 1539 pages from more than 600 authors from the dataset of IAM (Sentence Catalog). IAM database is a complex database with different data types. The selected collection contains documents grouped by author as shown in Figure 2. Use the SoftMax function to train low-resolution images. The image below shows an image of sentences from IAM dataset.

Preprocessing: The article is split into image sets of size 113×113. For each sentence text blocks of these sizes are created. Additionally, the generator function is used to generate random patches. Then they were converted to numpy arrays as shown in Figure 3.

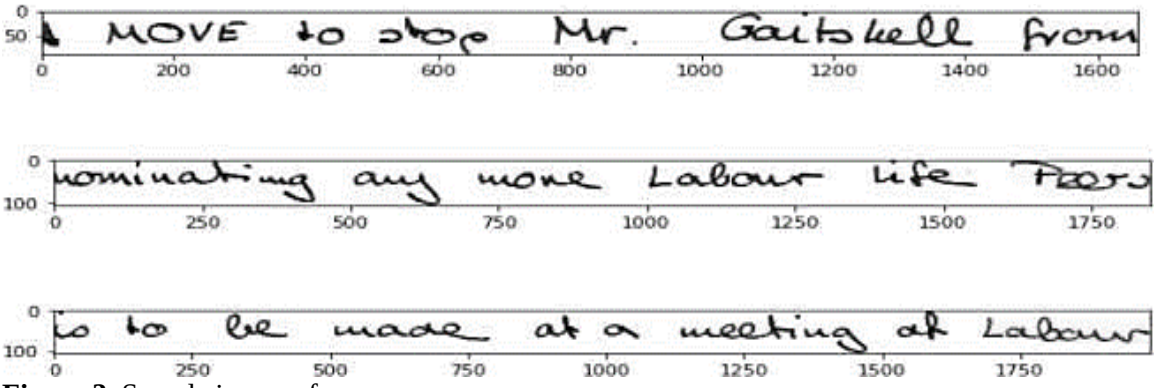

Figure 3. Sample image of numpy array.

Label encoders are used to encode labels into numbers. Figure 4 shows an example of label encoding done for the project. The label encoder encodes the labels within the range of 0 and n classes 1, where 'n' is the number of distinct labels as shown in Figure 4.

```
['../input/data_subset/data_subset/a01-000u-s00-00.png'
 '../input/data_subset/data_subset/a01-000u-s00-01.png'
 '../input/data_subset/data_subset/a01-000u-s00-02.png'
 '../input/data_subset/data_subset/a01-000u-s00-03.png'
 '../input/data_subset/data_subset/a01-000u-s01-00.png'] ['000' '000' '000' '000' '000'] [0
0 0 0 0]
```

Figure 4. Sample from Label Encoding.

Categories: CNN models built in python using TensorFlow libraries and Keras. The model was built by various convolutions using ReLU activation function of thick layers and convolutions. Start creating a distribution system using SoftMax.

Extraction from Features: In the feature extraction process of convolution and pooling of different sizes, letter spacing, curvature and other features of different writers are extracted. Feature extraction is done through multiple convolutions with maxpool layers. This project uses zero padding.

Training: Model training was done as follows:

- *Stage 1:* The filters were initialized with weights or parameters of random values
- *Step 2:* The training image input underwent the operations of convolution, ReLU and pooling during the forward propagation with the fully connected layer producing output probabilities. The output probabilities for the first training would be random because the weight was also randomly assigned.
- *Step 3:* The total error was calculated for the output layer
- *Step 4:* Backpropagation was used to estimate the gradients of errors and updated the filter and the weight parameters to minimize the error. The network learned to classify the image correctly

by altering the weights. The filter size and the number of filters got fixed and so did the architecture of the network.

- *Step 5:* Steps 2 to 4 were repeated for all the images. The parameters now have been optimized to appropriately classify images.

RESULTS AND DISCUSSION

The Adam optimizer uses the adaptive learning curve to enhance the pattern. The algorithm dynamically alters the training data of every parameter to increase the training convergence and efficiency of the model. After preliminary testing with 3 convolutional layers and 2 fully- connected layers, the model accuracy reached 84%. The model was brought up by examining each process and all combined processes and cycles until and unless the required results were achieved. The true advancement of the model is shown in Figure 5. More eminently, the training accuracy is 84.27% in the 1st period and remains more than 93% in the next period. Recognition accuracy peaked at 96.61% in the 4th round and then gradually decreased. The lowest rate in the second quarter was 89.39%. Finally, the model demonstrates the capability to update missing data, achieving 92% accuracy. This performance measure shows how well the model classifies the input image correct as shown in Figure 5 as shown in Table 1.

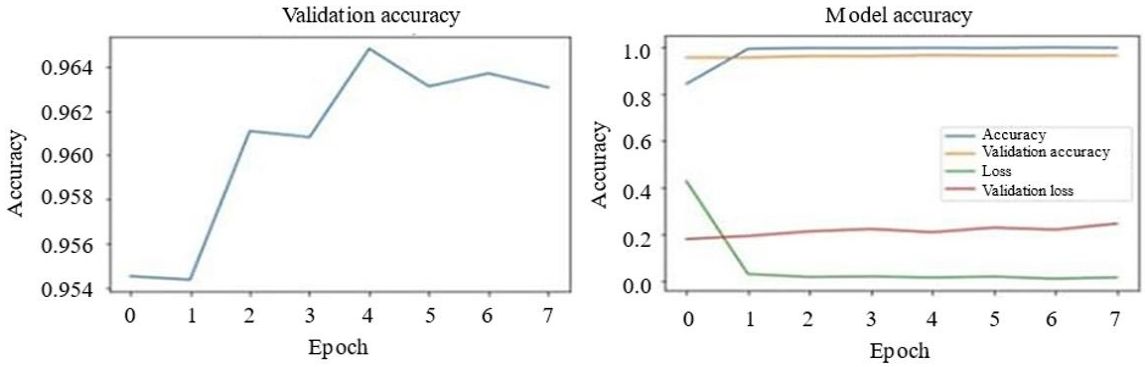

Figure 5. Performance of CNN Model.

Table 1. Analysis of the performance of the model.

Model	Preprocessing techniques	Main features of the model	Training Accuracy	Validation Accuracy	Test Accuracy
CNN Model	Slant correction and normalization	3 convolutional layers and Adam optimizer	97%	95.5%	92.6%

Future Works

First, we can use advanced techniques such as dithering for additional and consistent training. We can normalize the data by dividing every pixel by its standard deviation. Then we limited the training sample to 25 for each given word in order to test and update our model, given time and budget constraints. Alternative way to improve your attitude is to think beyond the solution. We will solve this problem by considering detailed but still useful decision algorithms such as beam search. We can use characters/words as language, add a penalty/point to each candidate's last search, and create a mixed softmax representing the characters/words. If the model shows the best candidate according to the softmax method and the search engine scores differently than other candidates, our model can adjust its behavior accordingly. The future development of applications based on machine learning and deep learning algorithms are unlimited. In the future, we will learn more intensive or hybrid algorithms than existing algorithms and use more information to complete different solutions. The use of algorithms in the future will differ between populations and schools, and this will be determined by the difference between algorithms described above. It can also be used in cases where we know from previous development that the application works very well. For example, from government secrets or secrets.

For example, you can use these algorithms, clinical trials, fingerprint xoo and retina scanners, files, data filters and countries, military and terrorist activities, etc. We may use it in clinical practice to improve the diagnosis, treatment and evaluation of patients, including search tools for tracking. Other problems in large and small groups. Advances in this field can help us create a safer, smarter and more efficient environment by using these algorithms in both routine and advanced applications in business or the public. Application-based artificial intelligence and deep learning are the future of the technology world because they can provide accurate and effective solutions to big problems.

CONCLUSION

The evaluation of the performance of the model is shown in the table. This model can be trained without the need for fitting, and the training accuracy is almost close to real use. Therefore, CNN alone can guarantee the accuracy of the text. Many tile removals and joint treatments are required to eliminate important features in the structure. This work shows that CNNs can capture important features in large images. The drawback of this project is that it can only be performed on GPU platforms due to its high computation requirements. One potential for future work is to try to use CNN to identify malicious articles online and use CNN state models to identify content written by authors.

REFERENCES

1. Bezerra, B. Zanchettin, C. and de Andrade, V. (2012). A MDRNN-SVM Hybrid Model for Cursive Offline Handwriting Recognition. Artificial Neural Networks and Machine Learning – ICANN 2012, pp.246-254.
2. Khan, A. Sohail, A. Zahoora, U. and Qureshi, A. (2020). A survey of the recent architectures of deep convolutional neural networks. Artificial Intelligence Review.doi: 10.1007/s10462-020-09825-6
3. Lee, H. and Verma, B., (2012) Binary segmentation algorithm for English cursive handwriting recognition. Pattern Recognition, 454, pp.1306–1317.
4. Putra, M.E.W. and Supriana, I., (2015) Structural offline handwriting character recognition using levenshtein distance. Proceedings - 5th International Conference on Electrical Engineering and Informatics: Bridging the Knowledge between Academic, Industry, and Community, ICEEI 2015, August, pp.31–36.
5. Rahman, A. and Verma, B., (2013) Effect of ensemble classifier composition on offline cursive character recognition. Information Processing and Management, 494, pp.852–864.
6. Fitrianingsih, F., Madenda, S., Ernastuti, E., Widodo, S. and Rodiah, R., (2017) Cursive Handwriting Segmentation using Ideal Distance Approach. International Journal of Electrical and Computer Engineering (IJECE), 75, p.2863.
7. Dasgupta, J., Bhattacharya, K. and Chanda, B., (2016) A holistic approach for Off-line handwritten cursive word recognition using directional feature based on Arnold transform. Pattern Recognition Letters, [online] 79, pp.73–79. Available at: http://dx.doi.org/10.1016/j.patrec.2016.05.017.
8. Birnberg, J.G., (2011) A Proposed Framework for Behavioral. 231, pp.1–43.
9. Hepzi, J.L., I.Muthumani and S.Selvabharathi, (2017) English Cursive Hand written Character Recognition. 117, pp.57–63.
10. Yann LeCun, Leon Bottou, Yoshua Bengio, P.H., (1998) A B7CEDGF HIB7PRQTSUDGQICWVYX HIB edCdSISIXvg5r ` CdQTw XvefCdS. PROC. of The IEEE. [online] Available at: http://ieeexplore.ieee.org/document/726791/#full-text-section.
11. Gonzalez, T.F., (2012) Handbook of approximation algorithms and metaheuristics. Handbook of Approximation Algorithms and Metaheuristics, pp.1– 1432.
12. Simonyan, K. and Zisserman, A., (2015) Very deep convolutional networks for large-scale image recognition. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, pp.1–14.

