

DOCUMENT IMAGE ENHANCER AND TEXT GENERATOR

[1] Guide – Mrs. N. Swapna Goud (Assistant Professor)
[2] B. Bhuvaneshwari [3] Gayatri Deepthi Y. [4] K. Keerthana

Department of CSE
ANURAG GROUP OF INSTITUTIONS

Abstract - A document may deteriorate in various ways over time, rendering it illegible. An efficient technological tool is required to make the paper legible again and remove noise. To address this issue, we employ an autoencoder based on convolutional neural networks (CNN). Its application can benefit machine-printed papers. You can convert a variety of papers into text data using optical character recognition (OCR), including scanned paper documents, PDF files, and camera-captured images. It is also used to convert document images into text data that can be read and analyzed by other applications. This data can be manipulated and used for data analytics, process automation, and efficiency enhancement.

Index Terms - Deteriorate, Noise, Illegible, Legible, Autoencoder, Conventional Neural Networks, Optical Character Recognition, Machine-printed papers, Process automation, Data analytics, Efficiency enhancement.

I. INTRODUCTION

Image denoising, whose basic aim is to reconstruct the original image by suppressing noise from a noise-contaminated version of the image, is one of the fundamental problems in the field of image processing and computer vision. Different intrinsic (sensor-related) and extrinsic (environment-related) conditions, which are frequently impossible to prevent in real-world applications, can result in image noise. Due to the importance of getting the original image content for effective performance, image denoising is used in a variety of applications including restoration of images, visual tracking, image registration, segmentation of images, and classification of images. A text image is transformed into a machine-readable text format through a procedure called optical character recognition (OCR). Your computer will save the scan as an image file, for instance, if you scan a document or a receipt. The words in the image file cannot be edited, searched for, or counted using a text tool. The image can be transformed into a text document with its information saved as text data using OCR, though. Data can be extracted from scanned documents, camera images, and image-only PDFs using OCR software. It removes the need for manually entering data by making static content editable.

II. EXISTING SYSTEM

The existing system utilizes algorithms such as Support Vector Machine (SVM), Block matching 3D (BM3D) and Weighted Nuclear Norm Minimization (WNNM). The primary drawback of Support Vector Machine (SVM) is how sensitive it is to the state of the documents. Problems can arise when there is a complicated image background or an uneven intensity. Block-matching 3D (BM3D) and Weighted Nuclear Norm Minimization (WNNM) have restricted applications in real-world problems as a result of the complexity and extensive patch matching required for their optimization. When there are many different image structures, as there often are in real-world situations, the manually and heuristically determined parameters are also rigid.

Disadvantages of existing system:

1. Their use in real-world applications has been constrained by their complicated optimization and thorough patch matching.
2. The manually and heuristically determined parameters are also rigid in situations where there are numerous distinct image structures, which is frequently the case in real-world scenarios
3. It does not extract text from images.

III. PROPOSED SYSTEM

The suggested system solves the issue by digitalizing and restoring damaged documents. It utilizes a Convolutional Neural Networks based Autoencoder to improve the input document picture by erasing noises like stains and folding lines, among other things. The improved image is then fed into an Optical Character Recognition (OCR) engine that outputs text using the tesseract Optical Character Recognition (OCR) in order to extract text from the document. An intuitive user interface makes it simple for users to submit the image of the desired document and receive the cleaned image and text that can be copied and restored in the future.

Advantages of proposed system:

1. It is user-friendly
2. It is applicable in real-world scenarios
3. It generates text from images

IV. METHODOLOGY

The proposed solution takes care of the problem by digitizing and repairing damaged papers. It uses a Convolutional Neural Networks-based Autoencoder to improve the image of the input paper by, among other things, removing noises like stains and folding lines. The enhanced image is then passed into an optical character recognition (OCR) engine that generates text using the tesseract OCR. Users can easily submit an image of the desired document and obtain the cleaned image and text that can be duplicated and restored later thanks to an easy-to-use user interface.

Below are the modules used in the proposed system.

1. Image Data Preparation

The images are loaded in grayscale and converted into arrays. Converting images into grayscale facilitates the simplification of methods and removes the difficulties associated with computational requirements. For those who are unfamiliar with image processing, it allows for less challenging learning. This is because grayscale compression reduces a picture to its most basic pixel. No matter what dimension, shape, or other characteristics numerical arrays have, you can perform lightning-fast actions on them using Numpy. Sometimes, image processing tools (OpenCV, PIL, scikit-image) wrap images in a unique format that utilizes Numpy in the background. The images can be specifically converted to Numpy arrays if Numpy is not already being used in the background. You can then quickly perform numerical computations on them.

2. Initialization

An Autoencoder class is created. It contains various methods to initialize encoder and decoder using many layers and activation functions, including Conv2D, BatchNormalization, MaxPooling2D, UpSampling2D, ReLU and Sigmoid. The Autoencoder-Convolutional Neural Networks model is designed to denoise noisy images. The Encoder consists of Conv2D and MaxPooling2D layers. The decoder consists of Conv2D and UpSampling2D layers. In order to remove the extra noise, the model learns a vector field for mapping the incoming data towards a lower dimensional manifold that describes the natural data. The encoder will be able to extract the most crucial features this way and acquire a more reliable representation of the data.

3. Validation

Validation is done on test data in order to determine how well your model has been trained, which depends on the quantity of your data, the value you want to predict, input, (and other factors), and in order to assess model properties, such as mean error, classification errors, precision, and recall for IR-models (and so on). It is carried out after model training. Autoencoder-CNN model is validated on a series of test images to make accurate predictions. The “eval_model()” method is called on the Autoencoder object and passes the test data as an argument. “self.autoencoder_model.predict()” in “eval_model()” is used to create a model and fit it with trained data to make a prediction.

4. Result

The Autoencoder-CNN model after learning noise from training images reconstructs the original image without the noise. A noisy or degraded image taken as input and noise is removed from it to generate an enhanced or denoised image as output using the Autoencoder-CNN model. The “im_show()” method is used to display the enhanced image, i.e., our output. It takes the predicted enhanced image and cmap as argument. To generate text from the enhanced image we utilize Pytesseract. It serves as a wrapper for the Tesseract-OCR tool. Denoising an image makes it easier to extract text from an image. The “pytesseract.image_to_string()” method is used to return the extracted text from an image as string and is then displayed as output.

V. EXPERIMENTAL TOOLS

Flask

Python-based microweb platform is Flask. Due to the lack of required tools or modules, it is classified as a micro-framework. It lacks any elements, such as a database abstraction layer, form validation, or other elements, where pre-existing third-party libraries already provide common jobs. Flask, on the other hand, permits extensions that can be used to add application functions in the same way that they were included in the original Flask code. In addition to utilities connected to popular frameworks, there are extensions for object-relational mappers, form validation, upload handling, and several open authentication systems.

OS

The os module, a built-in Python module, offers a means of communicating with the operating system. It has a wide range of functions for interacting with the system environment, controlling processes, and managing files and folders. The following are some examples of how to utilize the OS module: How files and directories work In the OS module, there are options for renaming, moving, deleting, and adding files and directories. Management of processes Tools for starting, pausing, and monitoring processes are provided by the operating system module.

Environment Variables:

The os module provides functions for getting and setting environment variables, which are used to store information about the setup of the system. File permissions, which regulate who can see and alter files, can be set, and retrieved using os module calls. Information about the system is available through the os module, including the hostname, the operating system version, and the current working directory.

Using the Shell:

The os module provides ways to interact with the shell by executing commands and retrieving their outcomes. Overall, the os module is a robust and adaptable module that provides several tools for interacting with the operating system. It is frequently used by many Python apps, such as file managers and system tools.

Glob

To locate files and path names that match a specific pattern, use Python's glob module. The pattern rules of glob follow the standard Unix path extension rules. Furthermore, benchmarks indicate that it matches pathnames in directories more fast than previous approaches. We can utilize wildcards (“*”, “?” and “[ranges]”) with glob to simplify and facilitate path retrieval in addition to exact string search.

OpenCV

Real-time computer vision tasks are the main emphasis of Open CV (Open Source Computer Vision), an open-source collection of programming tools. It processes images and videos to discover patterns, recognise objects, and perform many other computer vision tasks. Here are a few examples of applications for OpenCV: OpenCV is capable of detecting a variety of objects in both still and moving images, such as features, people, and automobiles. Image processing: A number of image processing jobs, such as picture filtering, border detection, sharpening, and smoothing, may be carried out using OpenCV. You can use Open CV to extract details from pictures and videos, such as corners, edges, and lines.

Machine Learning:

Open CV can be used with methods from machine learning to build models for image classification, object recognition, and other computer vision applications. Applications for augmented reality that overlay digital data over real-world objects can be created using Open CV. Robotics: Open CV can be used to create computer vision systems for robots that perform tasks like navigation, object identification, gesture recognition, and age screening. Medical Imaging: Open CV can be used to analyse medical images and videos, such as X-rays, MRI scans, and ultrasound images, in order to look for patterns and make diagnoses. In general, Open CV is a useful tool for researchers and programmers working on computer vision and image processing projects.

TensorFlow

For differentiable programming and dataflow across a variety of tasks, a free and open-source software library called TensorFlow is used. Additionally, neural network applications for machine learning use this symbolic maths toolkit. Both research and manufacturing are done with it at Google. TensorFlow was developed for use by Google by the Google Brain team. The Apache 2.0 open-source license was applied when it was released on November 9th, 2015. TensorFlow can be used in the following scenarios: Description of the Image: TensorFlow allows for the creation and training of deep learning models for picture classification tasks like object recognition.

Keras

Keras is a Python-based deep learning API that is built on top of the TensorFlow machine learning platform. It was designed to speed up experimentation and provide an excellent developer experience. The purpose of Keras is to provide an unfair advantage to any developer seeking to publish ML-powered applications. Simple, but not excessively so, is "Keras." Developers' cognitive load is reduced by Keras, allowing you to focus on the important parts of the problem. Usability, prompt troubleshooting, elegant and concise code, maintainability, and deployability are all highly valued principles in Keras. via TFLite, TFServing, and TF.js. Adaptable — The progressive disclosure of complexity principle states that arbitrary complex workflows should be feasible via a direct path that expands on what you have already learned. Keras upholds this principle.

NumPy

A general-purpose array processing tool is NumPy. In addition to tools for dealing with these arrays, it offers a high-performance multidimensional array object. It serves as Python's foundational module for scientific computing.

It has a number of characteristics, notable ones among them being:

An effective N-dimensional array object Complex (broadcasting) functions

Software for combining C/C++ and Fortran code, Fourier transform, random number, and useful linear algebra features NumPy can be used as a productive multi-dimensional container of general data in addition to its apparent scientific applications. Because NumPy supports arbitrary data-type definitions, it can quickly and easily integrate with a broad range of databases.

Pandas

Using its potent data structures, Pandas, an open-source Python library, offers high-performance data transformation and analysis tools. Python was primarily used for data processing and munging. It did not make much of an impact on data processing. Pandas figured out the solution. Regardless of the source of the data load, we can complete the five standard stages of data processing and analysis using Pandas: prepare, manipulate, model, and analyse. Python and Pandas are used in a variety of academic and professional areas, such as finance, economics, statistics, analytics, etc.

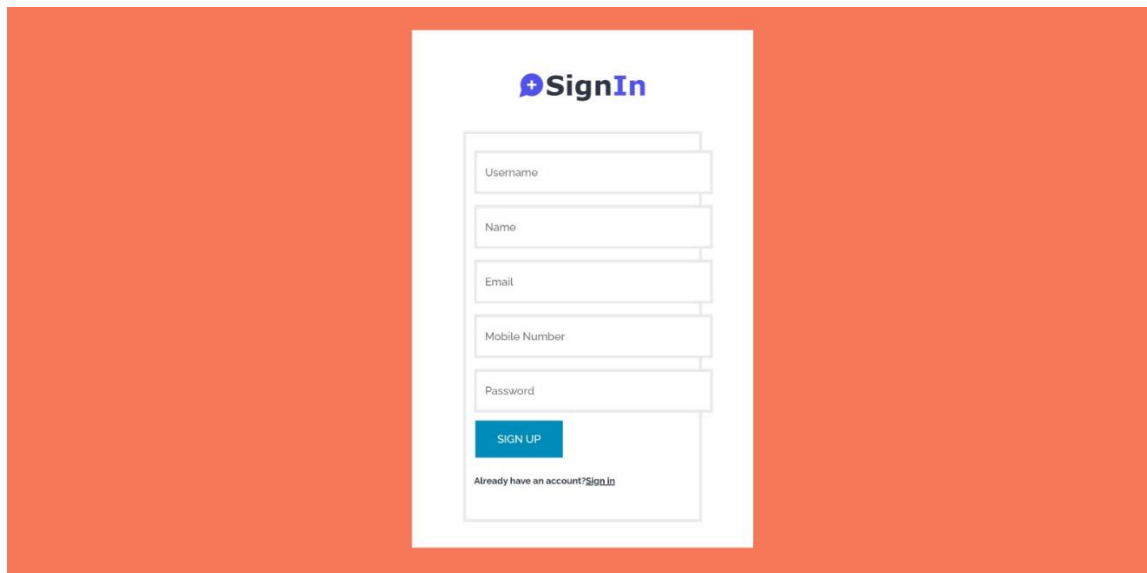
Matplotlib

A Python 2D plotting library called Matplotlib creates publication-quality graphics in a range of hardcopy forms and interactive settings across various devices. Four graphical user interface toolkits, the Python and IPython shells, the Jupyter Notebook, 38 web application servers, and Python programmers can all make use of Matplotlib. Matplotlib aims to make both simple and complex tasks possible. Plots, histograms, power spectra, bar charts, error charts, scatter plots, etc. can all be produced using just a few lines of code. Check out the sample plots and the thumbnail gallery for some samples. The Pyplot package offers a MATLAB-like interface for basic plotting, especially when used with Python.

Pytesseract

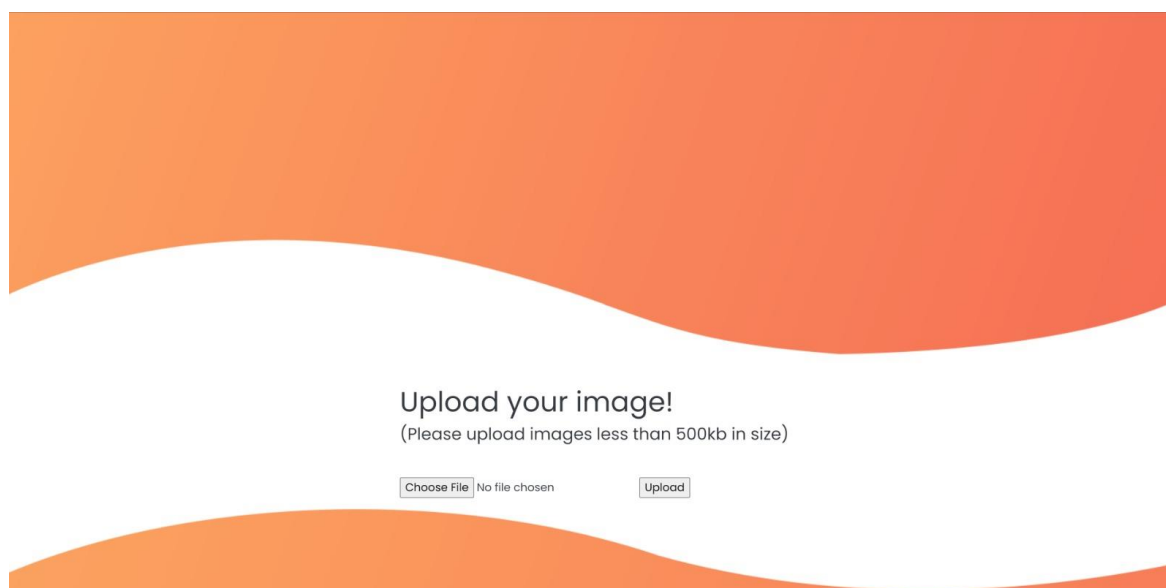
An Optical Character Recognition (OCR) programme written in Python is called Pytesseract, also known as Python-tesseract. It will scan and identify text on licence plates, in pictures, and other types of text. A class or module called Python-tesseract covers Google's Tesseract-OCR Engine. It can easily read all image types supported by the Pillow and Leptonica imaging libraries, which mainly include - jpeg, png, gif, bmp, tiff, and so on. This makes it useful as a standalone tesseract invocation script. Additionally Python-tesseract will show the recognised text when used as a script rather than writing it to a file.

VI. OUTPUT



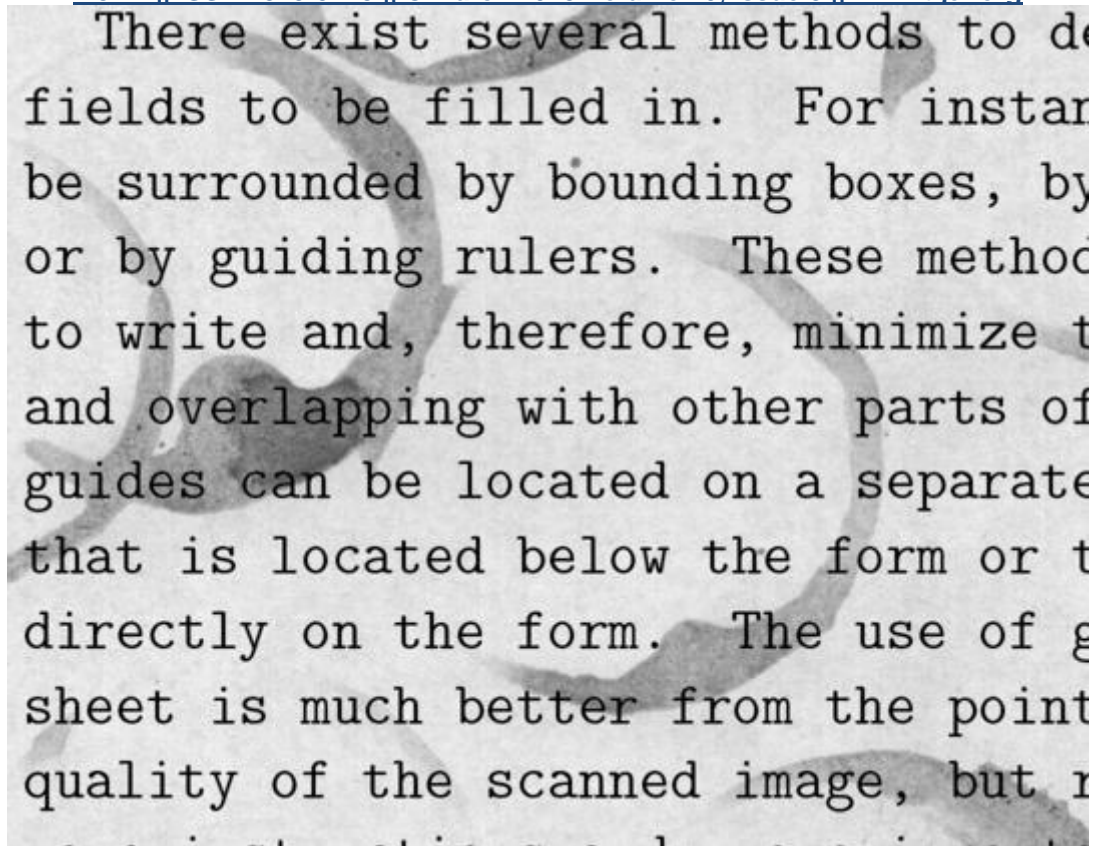
(Fig 1 Signup/login page)

The homepage of the user interface consists of the signup form. If the user is using the application for the first time, then the user needs to signup, else the user can login through his/her username and password.



(Fig 2 Input page)

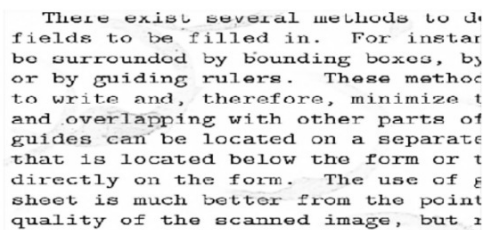
After the user logs in to the system, he/she is redirected to the input page which asks the user to upload an image to perform enhancement.



(Fig 3 Input image)

This is a sample input image that is uploaded to the system. It contains noises and needs to be denoised.

The result is:



For the given input image the is : **There exist several welhods to dt fields to be filled in. For instar be surrounded by bounding boxes, by er by guiding rulers. These methoc to write and, therefore, minimize t and overlapping with other parts of guides can be located on a separate that is located below the form or t directly on the form. The use of c sheet is much better from the point quality of the scanned image, but x**

(Fig 4 Output image and text)

The above figure consists of the enhanced image and the text extracted from it.

VII. CONCLUSIONS

Despite the rise of digitalization, documents are still what most people around the world continue to use. We developed a tool that utilizes Autoencoder built on Convolutional Neural Networks (CNN) and Optical Character Recognition (OCR). It helps in removing any type of noise from the image. Our tool aims to restore, enhance and denoise degraded documents to preserve important documents and aid further digitalization. This tool can be used to not only denoise old documents but also documents that have been wrinkled, or stained, or simply to correct a mistake in a printed document that a user does not have a digital copy of with the help of Optical Character Recognition technology.

VIII. REFERENCES

- [1] M. Malfait and D. Roose, "Wavelet-based image denoising using a Markov random field a priori model," in *IEEE Transactions on Image Processing*, vol. 6, no. 4, pp. 549-565, April 1997, doi: 10.1109/83.563320.
- [2] Hyvärinen, Aapo, Patrik O. Hoyer and Erkki Oja. "Sparse Code Shrinkage: Denoising by Nonlinear Maximum Likelihood Estimation." NIPS (1998)
- [3] Fan, Lixin & Fan, Liying & Tan, Chew Lim. (2003). Wavelet Diffusion for Document Image Denoising. 1188-1192. 10.1109/ICDAR.2003.1227845.
- [4] A. Achim, P. Tsakalides and A. Bezerianos, "SAR image denoising via Bayesian wavelet shrinkage based on heavy-tailed modeling," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 8, pp. 1773-1784, Aug. 2003, doi: 10.1109/TGRS.2003.813488.
- [5] J. Portilla, V. Strela, M. J. Wainwright and E. P. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain," in *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1338-1351, Nov. 2003, doi: 10.1109/TIP.2003.818640.
- [6] B. van Ginneken and A. Mendrik, "Image Denoising with k-nearest Neighbor and Support Vector Regression," 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, China, 2006, pp. 603-606, doi: 10.1109/ICPR.2006.685.
- [7] H. Takeda, S. Farsiu and P. Milanfar, "Kernel Regression for Image Processing and Reconstruction," in *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 349-366, Feb. 2007, doi: 10.1109/TIP.2006.888330.
- [8] K. Dabov, A. Foi, V. Katkovnik and K. Egiazarian, "Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering," in *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080-2095, Aug. 2007, doi: 10.1109/TIP.2007.901238.
- [9] R. Smith, "An Overview of the Tesseract OCR Engine," Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), Curitiba, Brazil, 2007, pp. 629-633, doi: 10.1109/ICDAR.2007.4376991.
- [10] Manoj Kumar Singh, Uma Shanker Tiwary and Young-Hoon Kim, "An Adaptively Accelerated Lucy-Richardson Method for Image Deblurring", Hindawi Publishing Corporation EURASIP Journal on Advances in Signal Processing Volume 2008, Article ID 365021, 10 pages, doi:10.1155/2008/365021
- [11] S. Esakkirajan, T. Veerakumar, A. N. Subramanyam and C. H. PremChand, "Removal of High Density Salt and Pepper Noise Through Modified Decision Based Unsymmetric Trimmed Median Filter," in *IEEE Signal Processing Letters*, vol. 18, no. 5, pp. 287-290, May 2011, doi: 10.1109/LSP.2011.2122333.
- [12] S. Gu, L. Zhang, W. Zuo and X. Feng, "Weighted Nuclear Norm Minimization with Application to Image Denoising," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 2862-2869, doi: 10.1109/CVPR.2014.366.
- [13] J. Xu, L. Zhang, W. Zuo, D. Zhang and X. Feng, "Patch Group Based Nonlocal SelfSimilarity Prior Learning for Image Denoising," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 244-252, doi: 10.1109/ICCV.2015.36.
- [14] O'Shea, Keiron & Nash, Ryan. (2015). An Introduction to Convolutional Neural Networks. ArXiv e-prints.
- [15] Michalak, Hubert & Okarma, Krzysztof. (2019). Improvement of Image Binarization Methods Using Image Preprocessing with Local Entropy Filtering for Alphanumeric Character Recognition Purposes. *Entropy*. 21. 562. 10.3390/e21060562.