[1]Suresh Dodda,

[2]Naveen Kunchakuri,

[3]Anoop Kumar,

[4]Sukender Reddy Mallreddy

Automated Text Recognition and Segmentation for Historic Map Vectorization: A Mask R-CNN and UNet Approach

Journal of Electrical Systems

Abstract: - Historic maps are essential for comprehending how buildings and landscapes have changed over time. For this—vectorization can be a useful method of analysis for an extensive collection of these maps. However, text overlaps with structural elements—often makes this process more difficult. Therefore, an automated pipeline for text recognition, pixel-level text mask creation, dataset generation, and text bounding box detection has been proposed. Findings shows—text segmentation, detection, and recognition were demonstrated by the combination of Mask Region-based Convolutional Neural Network (Mask R-CNN) and UNet model achieved a 99.12% of all text occurrences in images—which also attained an accuracy of 87.72% while collecting text inside bounding boxes. This end-to-end pipeline shows potential for a wide range of future uses, especially when it comes to text removal for the purpose of making historic maps easier to vectorize and analyze—which will improve the understanding of historical buildings and landscapes.

Keywords: Maps, Mask R-CNN, Vectorization, UNet

1. Introduction

Historic maps are priceless documentation of the changes in landscapes throughout time—offering vital information for a variety of disciplines such as civil engineering, urban planning, and historical studies. These maps provide a thorough explanation of how land use has changed throughout time by tracking the changes brought by natural disasters, disputes, and construction. Historic maps are especially valuable in the fields of civil engineering and construction because they help identify strong building foundations by highlighting geotechnical risks like abandoned quarries and mines. However, re-landscaping—a critical process in contemporary construction—is highly dependent on the precise data that historic maps offer—highlighting the financial benefits of digitizing and automating feature recognition in historical maps. The digitalization and automatic feature recognition of historical maps provide significant societal benefits that go beyond commercial ones. For instance, the University College London project—Legacies of British Slave-ownership—which compiled information on slave plantation estate in the British Caribbean between 1763 and 1833. Researchers tracked Jamaican sugar estates using geo-referenced maps from The National Library of Scotland's vast collection, providing insight into the history and effects of slavery. These initiatives highlight how crucial historic maps are to improve society's comprehension of major historical occurrences throughout the world.

However, in order to analyze and comprehend historical maps, raster images—which are made up of pixels—need to be converted into vector images, which are made up of points, lines, and curves. To automatically quantify spatial changes across time—vectorization improves the precision and effectiveness of historical analysis. However, the presence of text labels covering important features poses a typical obstacle to vectorizing historic maps and impedes precise feature extraction. Therefore, in order to solve this, text localization and extraction using Deep Learning (DL) techniques is done with the goal of getting overcome the drawbacks noted in earlier research, like that done by [1]–[4]. An automated pipeline for the creation of datasets, the detection of text bounding boxes, the development of pixel-level text masks, and text recognition was proposed in this study. By minimizing image modifications and maintaining the integrity of the underlying features, this method attempted to remove text from map images selectively, which would speed up the vectorization process. The National

[1] Independent Researcher, Atlanta, Georgia, USA, Email: sureshr.dodda@gmail.com

[2] Independent Researcher, Clarksburg, MD, USA, Email: Knav18@gmail.com

[3] Independent Researcher, CA, USA, Email: Anoop.kumar.2612@gmail.com

[4]Independent Researcher, Prosper, TX, USA, Email: sukender@ieee.org

Library of Scotland donated the OS 25-inch maps from 1892–1914, which were used in this study's historic maps of Edinburgh. These detailed and expansive maps included georeferenced metadata—which provides accurate locational context and increases the maps' usefulness for digital mapping and historical research. Therefore, in order to improve the training efficacy of DL models, the study's methodology required creating a custom dataset that replicated the text properties in the historical maps. Text detection and segmentation showed positive outcomes when Mask Region-based Convolutional Neural Network (Mask R-CNN) and UNet models were used. While the UNet model offered greater mask accuracy—which is necessary for text removal and in-depth feature analysis—the Mask R-CNN model performed exceptionally well in identifying text instances and enabling Optical Character Recognition (OCR). To maximize performance on the big and detailed map images, training these models needed careful consideration of aspects including learning rates and loss functions. The study also looked into ways to improve text detection accuracy and deal with the problems of huge, overlapping text, such as sliding windows and image rotation. To enhance the models' capabilities, more dataset enhancements were taken into consideration, like changing the text size and font variants.

The paper is as follows; in the following section, we'll look at the study's backdrop. The related works are presented in Section 3. The materials, procedures, data, and model are covered in Section 4. The experimental analysis is described in Section 5, and the study is concluded with some conclusions and ideas for future work in Section 6.

2. Background

Digital images are essential to computer-based analysis and storage because they use a grid of pixels to represent visual material—with each pixel including a color intensity value. These images usually go through pre-processing to improve analysis accuracy and standardise dataset, especially when they are historical or detailed like maps. In order to prepare images for further analysis by DL models—pre-processing could include noise reduction, color simplification, or channel adjustments. In essence, a digital image is a matrix, with dimensions that match its width (W), height (H), and color channels (C), creating a matrix that is $W \times H \times C$. A wide spectrum of colors is possible with 24-bit Red Green Blue (RGB) color—which supports over 16 million distinct colors. In RGB images, red, green, and blue are represented by three color channels, with pixel intensity values ranging from 0 (no intensity) to 2^{k-1} (full intensity). There are several formats used to store digital images, each with unique properties. For instance, Tag Image File Format (TIFF) files, which are often used for historical maps, preserve uncompressed image data and contain geo-location metadata, which is essential for mapping applications. Other formats, such as Portable Network Graphics (PNG) and Graphics Interchange Format (GIF)—which can handle 16.7 million colors but GIF is limited to 256, provide lossless compression. Joint Photographic Experts Group (JPEG), on the other hand, is appropriate for web use when file size and download speed are important factors since it uses lossy compression to minimize file size at the expense of some detail. However, by reducing the amount of colors in an image—color quantization makes the image data easier and often helps with noise reduction and feature recognition. That's why—many colors on historical maps are slightly different from one another due to ageing, printing, or digital scanning procedures. In color quantization, K-means clustering—an unsupervised Machine Learning (ML) technique—is frequently utilized. It uses color similarity to cluster pixels together, iteratively fine-tuning group centroids until stability. This technique supports color palette reduction and subsequent analysis activities by assisting in the identification of the image's dominant colors.

Another pre-processing step that is helpful for separating distinct features from backgrounds is binarization, or turning images into black and white. Images are reduced to binary values, where pixels that are above a predetermined threshold are set to one (white) and those that are below to zero (black). This can concentrate on the important components of an image and significantly reduce the file size. In color images, grayscale conversion comes before binarization—which is the process of combining RGB data into a single intensity value per pixel to represent the brightness of the image. There are several ways to convert between these two formats—the intensity approach—which averages RGB values, and the luminosity method—which uses weighted averages to better represent human vision. In binarization, thresholding is defining a pixel value cut-off to distinguish foreground from background. By determining a threshold that minimizes intra-class variation in the histogram of the image, Otsu's[2] approach automatically makes this distinction between the two. Details can be lost in digital images due to noise, which can often be caused by flaws in the sensor or outside influences. To smooth out these abnormalities, filters such as median filters and Gaussian blur filters are utilized. With Gaussian blur, high-frequency noise is reduced by averaging the values of nearby pixels using a kernel shaped like a bell—with median filters, noise is reduced without affecting edges because each pixel is replaced with the median value of its neighborhood. These image processing methods—from noise reduction to color quantization—are essential for getting digital images ready for in-depth examination. Such pre-processing procedures are essential for precisely capturing and evaluating the map's features during the digitization of historical maps. They make it possible to see details more clearly, make it easier to extract features, and raise the general standard of the digital maps. Pre-processing improves the usefulness of these maps in a range of analytical and research-focused applications while also helping to preserve historical data. These image pre-processing methods help analysts and researchers make more informed decisions in environmental studies, historical research, and urban planning by helping them understand the geographic and historical information included in maps. This emphasizes how crucial it is to pre-process digital images in the larger context of conserving and making use of historical cartographic records.

3. Related Works

In many different geospatial applications and analysis, texts—especially intersections—are essential. Historically, texts—which are considered point features—have helped with geo-referencing raster maps and geospatial dataset alignment. In addition, they have proven useful in the extraction of whole texts from raster maps. More recently, they have been applied in the fractal analysis of urban sprawl and the objective determination of the natural limits of cities [5]. With the introduction of Geographic Information Systems (GIS) [6]—vector road network datasets were widely available. But pre-GIS road network data, which are usually only seen on paper maps, present serious difficulties because GIS tools cannot analyze them until they have been scanned and processed. The relevance of automated techniques to transform physical maps into digital—vector-based representations for temporal and geographical analytics is highlighted by this digitization requirement according to [7]. Conventional methods of transforming data heavily rely on computer vision algorithms such as [8], [9]—which require specialized knowledge to determine the ideal settings. Furthermore, the conversion accuracy is compromised by the sometimes inadequate quality of ancient maps and the frequent overlap of graphical elements according to [10]. On the other hand, map conversion tasks have shown to benefit more from the use of DL methods, particularly Deep Neural Networks (DNN) such as [11], [12]. User participation in attribute selection is eliminated by DNNs, which autonomously determine the best attributes for distinguishing text from other map features. This feature increases the accuracy of removing text from physical maps and makes them easier to understand by non-experts. The extraction of geographical objects from Earth observation data is one of the object recognition tasks in which DNNs have been thoroughly studied and used such as [13], [14]. They are the best option for automatic text and other geospatial feature extraction from physical maps due to their excellent performance and versatility. [15] examines the use of Region-based CNN (RCNN), a DL technique, to locate text in scanned historical United States Geological Survey (USGS) maps of several U.S. cities. One new use in geospatial analysis is the extraction of text from physical maps using Deep CNN (DCNNs) such as [16], [17]. Although DCNNs have been used to identify several geographic elements in historical maps, such as railroads and human settlements, this study closes a research gap by focusing on text. The shift in text data analysis and digitization from conventional computer vision to DL, especially DCNNs, is a significant development. This change expands the possibilities for non-experts to participate in geospatial data analysis while also enabling more precise and effective processing of historical maps. DL has a wide range of possible uses in geospatial analysis, and as technology advances, so does its potential. These uses could lead to both useful and insightful study into how to comprehend and manage spatial situations.

4. Materials and Methods

CNNs are a particular kind of neural network that has transformed computer vision by effectively extracting local characteristics from images. CNNs use convolution layers to recognize and process local features, as opposed to

[2] Often employed in image segmentation applications, Otsu's method automatically generates suitable image thresholding levels by minimizing intra-class variance.

Artificial Neural Networks (ANNs), which learn pixel-to-label correlations [18]–[30]. This results in a significant reduction in memory consumption and training data needs. With [31] release of the first CNN model in the 1990s—which showed successful handwritten character recognition—CNNs underwent a revolutionary transformation. This achievement demonstrated how well CNNs extract complicated features from simple inputs more effectively than ANNs. Although ANNs are capable of handling small image classification tasks, the prohibitive memory requirements of larger images make them less feasible. CNN architecture includes layers such as convolutional, activation, and pooling layers, which vary according to the particular job. Convolutional layers take feature maps from images and apply non-linearity by passing them through activation layers made up of trainable filters. Pooling layers help to avoid overfitting and lower computational effort by reducing spatial dimensions while maintaining important information. CNNs are trained to predict class labels or probabilities in the image classification domain. To determine class probabilities, training models often apply sigmoid or softmax activation functions in conjunction with cross-entropy loss. By locating objects within images—object localization expands classification. This entails producing bounding box coordinates in addition to class predictions using loss functions that are customised for bounding box regression and classification. The more difficult problem of object detection is locating several items inside an image, requiring the use of non-max suppression techniques to remove overlapping bounding boxes. More advanced techniques, such as Region Proposal Networks (RPN) in models like Faster R-CNN, which include object localization and detection, have developed from more conventional strategies like sliding windows. DL methods such as UNet and Mask R-CNN provide accurate object segmentation within images, going beyond simple detection. UNet predicts pixel-wise class labels using an encoder-decoder structure, which is specifically built for medical image segmentation. An extension of Faster R-CNN, Mask R-CNN uses Region of Interests (ROI) aligns to provide precise pixel mapping and provides both detection and instance segmentation functions. The procedure of creating ground truth masks is a labor-intensive but careful step in the training of segmentation models. Colour clustering is one technique that can simplify the color palette of images and speed up this process by making mask creation easier. DL algorithms like this are quite helpful for analysing historic maps. Through their ability to extract and segment text in great detail, they make it possible to digitise and analyze historical documents with greater accuracy. The tools required to convert pixel-rich images into structured, analyzable data are provided by CNNs, especially those made for segmentation. Moreover, CNN progress is a reflection of DL's increasing ability to handle complicated, high-dimensional data in a variety of domains. CNNs are now essential for driving meaningful information from visual data, which propels advances in academics and technology ranging from medical imaging to historical map research. The use of CNNs in historical map processing is a good example of how these networks support historical artefact research and digital preservation. In the domains of geography, history, and archival science, CNNs play a major role by automating the recognition and division of text and features on maps, transforming static images into dynamic information ready for investigation. CNNs are essential to modern computer vision, as seen by their progression from simple image classification to complex object recognition and segmentation. Their application in historical map analysis is a prime example of this union of technology and humanities.

4.1 Dataset Analysis

For the purpose of training object detection or segmentation models, a custom dataset must be created, particularly for specialized data types such as map texts. Therefore, a series of stages are involved in creating such a dataset, all aimed at producing a collection of images that closely resemble the intended application—in this case—map text recognition and segmentation. Firstly, the map images undergo preprocessing. A map section is subjected to a k-means clustering method with $k = 3$ in order to determine the color centroids. This stage modifies the colors of the map's pixels to correspond with these centroids, which helps to improve text recognition when training the model and unify the appearance of the text. This preprocessing reduces pixelation to increase character clarity while also bringing the synthetic text into line with the original map's style. The next step is to create map visuals without any text. To accomplish this, the maps are manually edited to erase any existing text using graphics software such as Photoshop or Affinity Designer. This clean surface makes it possible to add fresh, artificial text without any hindrance. Next, a thorough dictionary of fonts and texts is assembled. This dictionary is enhanced with fonts that resemble those in the original maps and consists of a combination of character sets, randomly generated numbers, and a list of significant words. This diversity guarantees that the dataset encompasses a wide range of text structures and styles, reflecting the variability found in real-world map data. Parts of the modified maps are used as backgrounds for text from the dictionary that has been randomly selected and placed in order to create dataset images. Every text segment is evaluated based on its bounding box and assigned an overlap score that represents the level of visual conflict with the features already present on the map. Lower scoring texts that show less overlap are favored in the dataset to preserve readability and prevent confusion. Following a scoring system that balances text clarity and overlap, the text proposals are sorted, and choices are made. In order to maximize text visibility and fit inside the map environment, this level was calculated through experimentation. To avoid overlapping text occurrences and provide distinct and simple data samples for model training, texts with non-intersecting bounding boxes are introduced to the dataset. Every chosen text is layered on the matching map section, and associated metadata is captured in Extensible Markup Language (XML) format, such as overlap scores, bounding box coordinates, and masks. The object detection or segmentation models require this structured data annotation in order to be trained and validated. The procedure is parallelized, using many Central Processing Unit (CPU) cores to generate distinct image samples simultaneously, to speed up the development of the dataset. The time needed to produce a large number of training and testing images is greatly decreased by this parallel processing, which improves the effectiveness of the dataset preparation stage. This strategy to create custom datasets for map text recognition includes a methodical preparation, text removal, development of synthetic text, and cautious selection based on overlap scoring. Accurate and reliable item detection and segmentation models can be trained using a well-constructed, customized dataset that closely reflects the intricacies and variances of map texts.

4.2 OCR Analysis

In operations like postal mail sorting and document digitization, OCR plays a critical role in transforming text from digital images into searchable and editable data, thereby saving a significant amount of manual labour. Structured and unstructured text images are the two main categories to which OCR algorithms are used. Text that has predictable elements, such standard fonts and clear layouts, and is often seen in books or official documents is referred to as structured text. Template matching is a common technique used in OCR algorithms for structured text. It compares a template image to a target image and finds matches based on a similarity function. The Sum of Squared Differences (SSD) and Sum of Absolute Differences (SAD) are common functions that compute the differences in pixel values between the target and template images. Zero-Mean Normalised Cross Correlation (ZNCC) is another method that shows how closely the two images are correlated. The Tesseract OCR engine is a well-known technology for processing structured text that was first created by HP Labs and then made open-source. Its efficacy and simplicity of usage have led to its popularity as an efficient way to translate visual text to strings. Preprocessing is necessary for best results because Tesseract's accuracy can be affected by image manipulations or different text orientations. Unstructured text, sometimes referred to as natural scene writing, is more difficult to work with because of its erratic orientation and arrangement. Therefore, traditional OCR techniques that are meant for structured text work less well. In unstructured scenes, advanced techniques such as CNNs are used to identify text bounding boxes. One such technique is the Efficient and Accurate Scene Text (EAST) detector, which recognizes text sections in images by applying a DNNs. Therefore, EAST follows a UNet-like design for merging feature maps, integrating features at several levels to accommodate heterogeneity in text size. In EAST, text places are predicted geometrically (using a rotatable box or quadrangle) and a score map is created throughout the detection phase. An Intersection-over-Union (IoU) loss for geometric alignment and a cross-entropy loss for accuracy are used to balance a loss function that combines score map and geometry losses to assess the performance of the model. OCR technology has developed to meet the challenges presented by both organized and unstructured texts, especially with advances in ML and DL. OCR's automation of text extraction greatly improves data processing and analysis efficiency by allowing for the quicker and more precise extraction of information from images. This development creates new opportunities for applications in a number of domains, such as real-time text translation, document archiving, and geographic analysis. The progression of OCR from simple template matching to complex DL models such as Tesseract and EAST signifies a significant development in text recognition technology. These advances make it easier to extract text from both organised and unstructured images, which is crucial for a variety of digital processing applications.

5. Experimental Analysis

5.1 Template Matching

OCR uses template matching as a technique to recognize particular symbols or patterns in an image. When working with text in different fonts, sizes, and orientations—as usually the case with maps—this procedure gets more complicated. Applying template matching to such a wide range of features without preprocessing would require an unmanageable amount of templates and result in a large number of False Positives (FP). On the other hand, consistent symbols such as trees and shrubs on maps, whose orientations are fixed and variances are minimal, are best identified using template matching. Researchers can determine the distribution of greenspace and monitor changes in it over time by comparing the positive matches of these symbols with historical maps. The initial stage of template matching involves collecting the templates, which is usually accomplished by resizing the symbols on the original map. Then, using a technique similar to zero-mean normalised cross-correlation, these templates are compared against the complete map. By balancing precise detection with a low number of FP, this method assists in identifying matches that surpass a predetermined threshold. For example, a threshold of 0.6 can be used to differentiate between actual trees and other features, such as text or buildings as shown in Fig. 1. An IoU threshold is used to remove redundant matches while maintaining nearby but separate trees when handling numerous detections of the same tree using non-max suppression as shown in Fig. 2. To maximize efficiency and minimize errors, this procedure necessitates meticulous parameter adjustment. The effectiveness of the template matching strategy was demonstrated in a case study of a heavily urban map of Leith, where it found many tree symbols with a low proportion of FP. False matches, which often involve text elements, emphasise how crucial text removal and other preprocessing steps are to enhance accuracy. One simple way to analyze greenspace density is to count template matches in each map pixel's designated neighbourhood. But this method gives each neighbourhood area the same weight, which could not precisely reflect how green space is actually distributed. The centre of the template match contributes more heavily to the greenspace score, decreasing with distance, in an advanced method that employs a weighting method based on a 2D Gaussian distribution as shown in Fig. 3. This approach can be modified in response to experimental or theoretical evidence, enabling a more nuanced evaluation of the concentration of green space. Finding the Euclidean distance between each pixel and the closest template match is another method for evaluating greenspace as shown in Fig. 4. This method clearly illustrates the distribution of greenspace and helps to identify unique subregions.

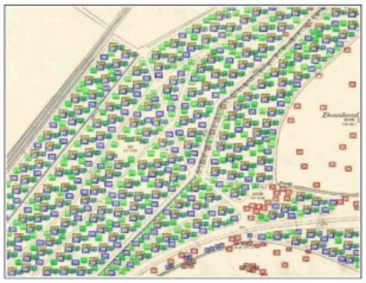

Fig. 1. A subset of the map is subjected to tree template matching with a threshold of 0.6; each box indicates a match, and the color of the box indicates which template it matches with

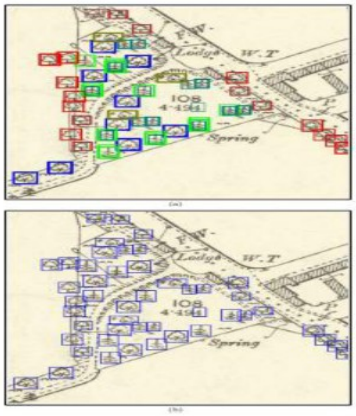

Fig. 2. (a) The ZNCC template matching method's final template matches with a 0.6 threshold. (b) With an IoU threshold of 0.7, the suppressed template matches employing non-maximum suppression

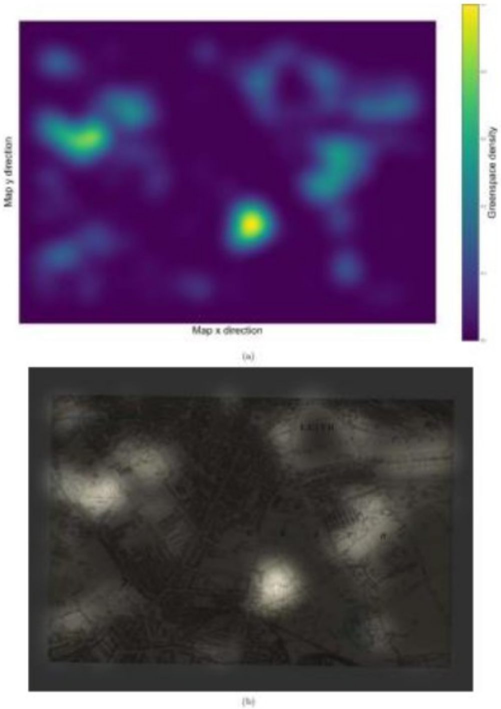

Fig. 3. Gaussian density plot (a) At each template center, a zero matrix matrix is multiplied by a Gaussian distribution with σ = 400. (b) The resulting Gaussian density plot overlay

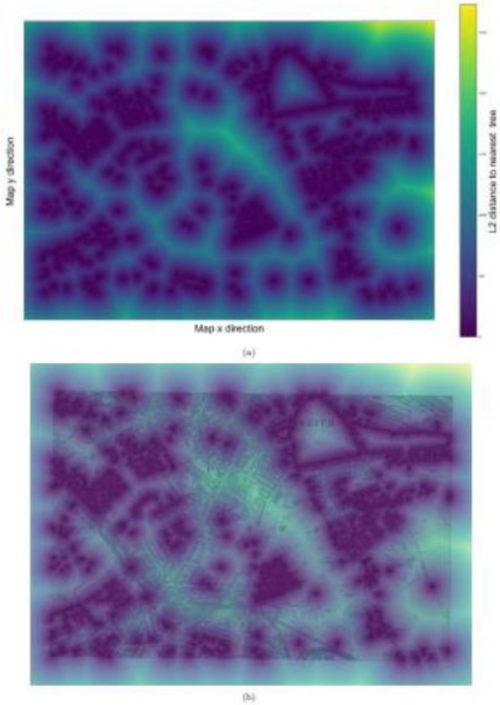

Fig. 4. (a) A plot in which the Euclidean distance between each pixel and the closest template center is represented by its value. (b) The original map image is superimposed with the Euclidean template distance plot

5.2 Pre-Trained Text Detection Models

We evaluated two DL models that had already been trained in order to establish a benchmark. Using the ICDAR[3] datasets from 2013 and 2015, OpenCV's EAST text detector was trained. Another example is CharNet, an advanced scene text detector that was also trained using the ICDAR dataset from 2015. As may be seen in Figs. 5 and 6, we applied CharNet to both the clustered and a binaryized image. Without anticipating any FP scenarios, the model precisely determines and detects some of the text in the images. CharNet clearly identifies a lot more text on the clustered image than on the binary image. In contrast, Fig. 7 shows that the EAST detector outperforms CharNet in terms of performance. EAST finds it difficult to distinguish the text from usual image noise caused by the buildings. As a result, it makes several incorrect region predictions.

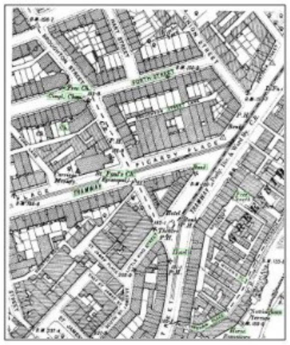

Fig. 5. CharNet was trained using a black and white map sub-image and the ICDAR 2015 dataset

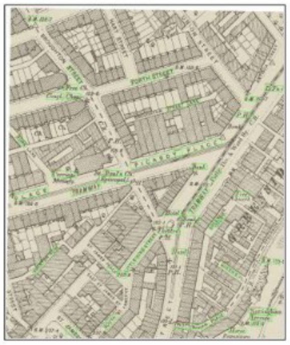

Fig. 6. CharNet applied to a $k = 3$ clustered map sub-image was trained using the ICDAR 2015 dataset

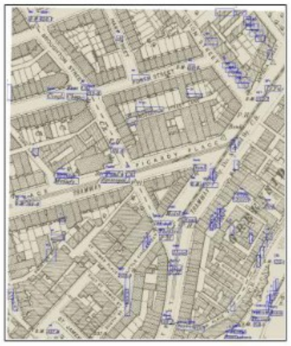

Fig. 7. Using a $k = 3$ clustered map sub-image, a pretrained EAST detector model

5.3 Mask R-CNN

[3] https://paperswithcode.com/dataset/icdar-2015

The twin capabilities of a Mask R-CNN model in object detection and instance segmentation made it the best choice for text recognition and segmentation on maps. This capability is very helpful for vectorizing map objects, such as buildings, when combined with traditional OCR techniques to recognize text. The Matterport version of Mask R-CNN, which combines a ResNet101 backbone with a Feature Pyramid Network—was initially trained on the COCO[4] dataset and was utilized by the model as shown in Fig. 8. Transfer learning was used to improve the convergence and efficiency of model training. This required modifying the trainable layers of the Mask R-CNN, including the mask generator, classifier, and RPNs, and initialising the network with pre-trained weights. Throughout 19 epochs, a collection of 1,000—1024 x 1024 map image segments were used for training, with a learning rate of 0.009 and a batch size of two as shown in Fig. 9. The RPNs used several anchor scales to recognize small text instances as shown in Fig. 10. To improve the model's convergence, especially in mask loss, the learning rate was lowered after a plateau in the loss function was noticed. As demonstrated by better mask generation and fewer occurrences of multiple predictions of text, this modification resulted in increased performance. Still, there were issues with the model's accuracy in masking longer words. This problem persisted despite attempts to adjust training durations and learning rates. A windowing approach was used to adapt the model to larger map visuals than the 1024 x 1024 samples for training as shown in Fig. 11. To achieve thorough text detection across boundaries, this required segmenting bigger images into overlapping windows. Non-maximum suppression was then applied to get rid of duplicate detections. To improve the model's capacity to handle a variety of text styles and orientations, additional model improvement involved changing the dataset to include text with different spacing and larger fonts. Even in complicated circumstances with noise and overlapping parts, the model's performance in text detection and segmentation increased after it was retrained with these improvements. With challenging images, the retrained Mask R-CNN proved its abilities by correctly recognizing and segmenting text among other map elements. This demonstrated that the model could adjust to the many textual features present in maps, facilitating the precise extraction and processing of textual data.

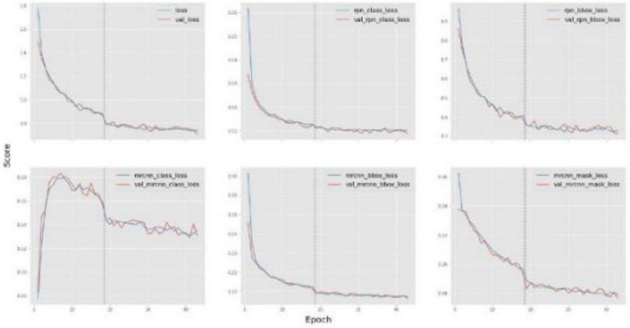

Fig. 8. Graphs showing the various loss functions for the Mask R-CNN model during training (top-left), the RPN (top-middle & top-right), and the Mask R-CNN prediction layers (bottom row)

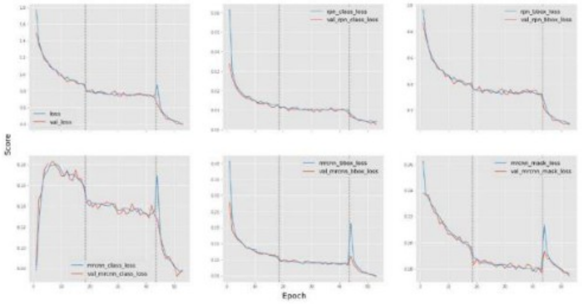

Fig. 9. Graphs showing the various loss functions for the Mask R-CNN model during training (top-left), the RPN (top-middle & top-right), and the Mask R-CNN prediction layers (bottom row) on 19 epochs

[4] https://cocodataset.org/

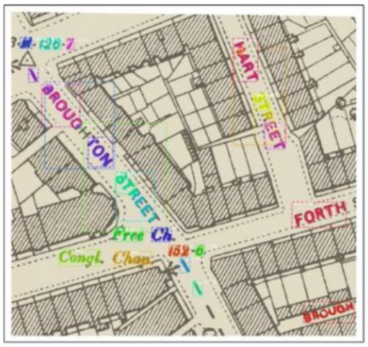

Fig. 10. A 1024 × 1024 K3 clustered unseen map image was fed into our Mask R-CNN model, which was trained solely on the RPN, Classifier, and Mask generator top layers

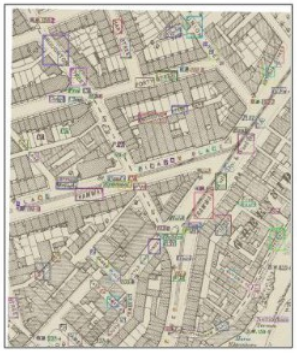

Fig. 11. The bounding boxes obtained from the sliding window approach, which applies the learned Mask R-CNN model over a bigger unseen image, prior to non-max suppression

5.3.1 Text Recognition with Mask R-CNN

Following a neural network's detection of the text's mask and bounding box—OCR is used to identify the text. For OCR to be effective, the text must be properly aligned and noise-free. Text alignment and noise reduction are made easier by using the predicted mask's information, which pinpoints individual character locations. One useful method for reorienting text is to use the minimal area rectangle that surrounds the mask. By providing the text's orientation, this rectangle enables the image to be rotated to align the text. Different rotation computations are needed depending on whether the text is inclined upward or downward. This approach works well for longer text strings since the orientation of the text is clearly indicated by the longest side of the rectangle as shown in Figs. 12 and 13. Nevertheless, this method has drawbacks when dealing with single characters or when the edges of the rectangle have comparable lengths, since it could mis predict the orientation of the text. Other approaches, such as pattern matching, would be more appropriate in these situations to establish the proper text alignment. After the text has been properly oriented, unnecessary noise surrounding the text can be eliminated using the mask and its bounding rectangle, improving the text's clarity for the OCR process. For accurate text recognition, particularly in complex map images where text may be surrounded by multiple graphical elements, this noise reduction is essential.

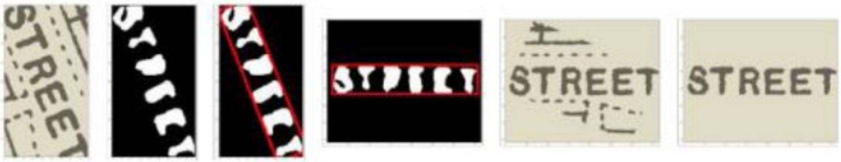

Fig. 12. The Mask R-CNN model predicted the reduced STREET text bounding box and mask, with the red box representing the minimal bounding rectangle

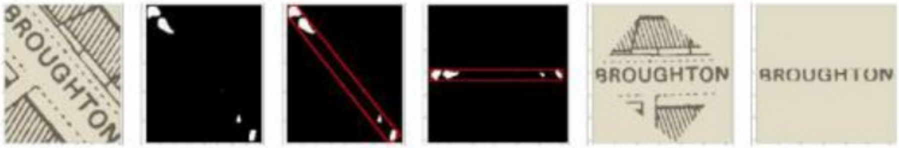

Fig. 13. The Mask R-CNN model predicted the clipped BROUGHTON text bounding box and mask, with the red box being the minimum bounding rectangle

5.4 UNet

The experiment showed that although the Mask R-CNN model can successfully identify text instances and provide enough mask details for OCR, it cannot completely remove text from images. To tackle this, the same custom dataset of 1024 x 1024 images was used to train a UNet semantic segmentation model, which was then optimised with a dice loss function and low learning rates to guarantee accurate convergence on bigger image sizes. The model's sensitivity to learning rate option meant that low learning rates were required in order to get reliable predictions on larger images. When it came to evaluating the finer elements of the text, the UNet model outperformed the Mask R-CNN in terms of producing precise pixel masks as shown in Fig. 14. Larger, overlapping text fonts presented challenges for the model, indicating that the dataset may need to be modified in order to improve the prediction power of the model. Sliding window approach was used to enhance the UNet model's performance and enable it to handle larger images more reliably as shown in Fig. 15. Furthermore, testing the model at different rotations improved its capability to identify characters that had previously escaped notice, but inadvertently obscured smaller fixed-orientation objects such as trees. This problem brought to light the possible advantages of adding a variety of image orientations to the training dataset in order to increase the accuracy and robustness of the model. To further address the difficulties in segmenting larger overlapping text, more changes were applied to the dataset. Wider overlap score ranges and fonts resembling those in problematic text sections were added to the dataset generator's parameters. Improved mask predictions for large text were obtained by training the UNet model on a redesigned dataset of smaller 512 x 512 images. This suggests that the model is now more capable of handling challenging text segmentation tasks. The significance of ongoing model and dataset optimisation in improving text segmentation performance was highlighted by these trials. The UNet model was able to handle the complexities of historic map text by optimizing the dataset and training procedure, especially when dealing with larger, overlapping text parts. The investigation process demonstrated that the Mask R-CNN and UNet models each have unique advantages when it comes to text segmentation and detection tasks. Mask R-CNN is good at finding text instances and enabling OCR, whereas UNet is better at text segmentation, which is important for text removal. For complex text processing tasks in historic map analysis, the two models' complimentary relationship provides a strong basis.

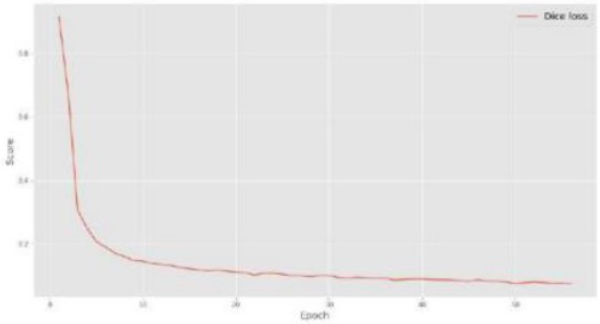

Fig. 14. A chart representing the dice loss function for the initial 56 training epochs of a UNet segmentation model

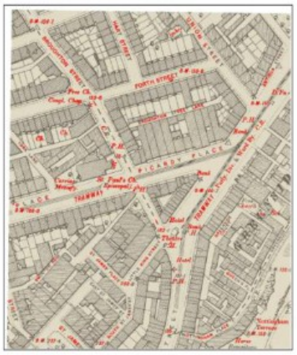

Fig. 15. Using a larger, unseen image and the trained UNet model's predictions, apply the sliding window approach

5.4.1 Text Recognition with UNet

The Mask R-CNN model showed limits in correctly predicting text masks, which are essential for appropriately orienting and denoising the text, during the examination of text detection and recognition on large images. This mask prediction error makes it more difficult to eliminate unnecessary visual noise surrounding text inside the minimum area rectangle. It was discovered that adding padding around this region generally would not be sufficient because it would unintentionally generate more noise and make text identification more difficult. In order to overcome these obstacles, an improved strategy was used, involving the use of a UNet model that was particularly trained for more accurate text character prediction. This model improvement made it possible to distinguish text within images more precisely, which enhanced text clarity and noise reduction. First, text regions are identified using the bounding box information from the Mask R-CNN model. These areas are then extracted from the more detailed mask created by the UNet model. Text segmentation is made much more precise by the incorporation of the UNet model. Through the use of the UNet's accurate masking capabilities and focusing on the regions defined by the bounding boxes of the Mask R-CNN, the resulting text masks show increased detail and accuracy as shown in Fig. 16. With this dual-model method, the issue of noise inclusion is substantially reduced and the important text elements are preserved for further OCR processing. The results of the comparative analysis show that using this combination approach is preferable to using the Mask R-CNN model alone. The combined masks from the Mask R-CNN and UNet models offer a more thorough and precise depiction of text in images, enabling improved text alignment and noise reduction.

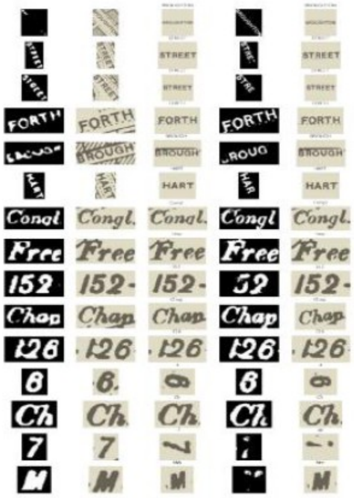

Fig. 16. The text masks as predicted by the trained UNet model are displayed in the fourth column. The fifth column displays the predicted images following orientation adjustments and noise reduction using the updated predicted masks in conjunction with the updated predicted text generated by Pytesseract

6. Conclusion and Future Works

In order to train DL models like as Mask R-CNN and UNet to recognize and segment text on historical maps, we created a special dataset for this study. The Mask R-CNN model accurately captured 87.72% of the text and identified 99.12% of text instances with high accuracy in text localization and mask prediction. The model provided enough information for text orientation and subsequent OCR processing, even if it had some issues with mask prediction for big images. By combining the UNet model with Mask R-CNN, the accuracy of character segmentation and noise reduction was improved. The models' synergy enhanced text recognition and made it easier to remove non-text components. The model's performance was further improved by making adjustments to the dataset, such as adding larger font sizes and changing the text spacing, especially in difficult situations where text overlapped with map features. In order to assess greenspace density, we also used a template matching technique in our approach, which showed a remarkable accuracy rate. In the future, the dataset could be further refined by overlaying tree symbols rather than text and training a CNN classifier to lower FPs in template matching. The potential for these approaches to assist in many applications, such as text removal and OCR integration, is demonstrated by the study's success in automating dataset generation and training efficient models for text recognition and segmentation on historic maps. In order to improve the restoration and analysis of historical maps, future research could explore advanced in painting techniques for text removal that maintain underlying map features.

References

[1] I. Iosifescu, A. Tsorlini, and L. Hurni, “Towards a comprehensive methodology for automatic vectorization of raster historical maps,” *e-Perimetron*, vol. 11, no. 2, pp. 57–76, 2016, Accessed: Apr. 11, 2024. [Online]. Available: www.e-perimetron.org

[2] N. Kamuni, H. Shah, S. Chintala, N. Kunchakuri, and S. A. O. Dominion, “Enhancing End-to-End Multi-Task Dialogue Systems: A Study on Intrinsic Motivation Reinforcement Learning Algorithms for Improved Training and Adaptability,” *2024 IEEE 18th International Conference on Semantic Computing (ICSC)*, pp. 335–340, Feb. 2024, doi: 10.1109/ICSC59802.2024.00063.

[3] N. Kamuni, I. G. A. Cruz, Y. Jaipalreddy, R. Kumar, and V. K. Pandey, “Fuzzy Intrusion Detection Method and Zero-Knowledge Authentication for Internet of Things Networks,” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 16s, pp. 289–296, Feb. 2024, Accessed: Mar. 31, 2024. [Online]. Available: https://ijisae.org/index.php/IJISAE/article/view/4821

[4] N. Kamuni, S. Chintala, N. Kunchakuri, J. S. A. Narasimharaju, and V. Kumar, “Advancing Audio Fingerprinting Accuracy Addressing Background Noise and Distortion Challenges,” *2024 IEEE 18th International Conference on Semantic Computing (ICSC)*, pp. 341–345, Feb. 2024, doi: 10.1109/ICSC59802.2024.00064.

[5] D. Ozturk, “Assessment of urban sprawl using Shannon’s entropy and fractal analysis: a case study of Atakum, Ilkadim and Canik (Samsun, Turkey),” *Journal of Environmental Engineering and Landscape Management*, vol. 25, no. 3, pp. 264–276, Jul. 2017, doi: 10.3846/16486897.2016.1233881.

[6] M. F. Goodchild, “Geographic information systems and science: Today and tomorrow,” *Annals of GIS*, vol. 15, no. 1, pp. 3–9, Nov. 2009, doi: 10.1080/19475680903250715.

[7] A. Crivellari and A. Ristea, “Crimevec-exploring spatial-temporal based vector representations of urban crime types and crime-related urban regions,” *ISPRS International Journal of Geo-Information*, vol. 10, no. 4, p. 210, Apr. 2021, doi: 10.3390/ijgi10040210.

[8] K. Malik, C. Robertson, S. A. Roberts, T. K. Remmel, and J. A. Long, “Computer vision models for comparing spatial patterns: understanding spatial scale,” *International Journal of Geographical Information Science*, vol. 37, no. 1. Taylor & Francis, pp. 1–35, 2023. doi: 10.1080/13658816.2022.2103562.

[9] W. Li and C. Y. Hsu, “GeoAI for Large-Scale Image Analysis and Machine Vision: Recent Progress of Artificial Intelligence in Geography,” *ISPRS International Journal of Geo-Information 2022, Vol. 11, Page 385*, vol. 11, no. 7, p. 385, Jul. 2022, doi: 10.3390/IJGI11070385.

[10] A. Tortora, D. Statuto, and P. Picuno, “Rural landscape planning through spatial modelling and image processing of historical maps,” *Land Use Policy*, vol. 42, pp. 71–82, Jan. 2015, doi: 10.1016/j.landusepol.2014.06.027.

[11] B. Huang, B. Zhao, and Y. Song, "Urban land-use mapping using a deep convolutional neural network with high spatial resolution multispectral remote sensing imagery," *Remote Sensing of Environment*, vol. 214, pp. 73–86, Sep. 2018, doi: 10.1016/j.rse.2018.04.050.

[12] S. Tarride, A. Lemaitre, B. Coüasnon, and S. Tardivel, "Combination of deep neural networks and logical rules for record segmentation in historical handwritten registers using few examples," *International Journal on Document Analysis and Recognition*, vol. 24, no. 1–2, pp. 77–96, Jun. 2021, doi: 10.1007/S10032-021-00362-8/FIGURES/13.

[13] S. Lang, G. J. Hay, A. Baraldi, D. Tiede, and T. Blaschke, "GEOBIA achievements and spatial opportunities in the era of big earth observation data," *ISPRS International Journal of Geo-Information*, vol. 8, no. 11. 2019. doi: 10.3390/ijgi8110474.

[14] W. Zhao, S. Du, and W. J. Emery, "Object-Based Convolutional Neural Network for High-Resolution Imagery Classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 7, pp. 3386–3396, 2017, doi: 10.1109/JSTARS.2017.2680324.

[15] A. E. Maxwell, W. E. Odom, C. M. Shobe, D. H. Doctor, M. S. Bester, and T. Ore, "Exploring the Influence of Input Feature Space on CNN-Based Geomorphic Feature Extraction From Digital Terrain Data," *Earth and Space Science*, vol. 10, no. 5, p. e2023EA002845, May 2023, doi: 10.1029/2023EA002845.

[16] J. Yue, W. Zhao, S. Mao, and H. Liu, "Spectral-spatial classification of hyperspectral images using deep convolutional neural networks," *Remote Sensing Letters*, vol. 6, no. 6, pp. 468–477, Jun. 2015, doi: 10.1080/2150704X.2015.1047045.

[17] Y. Z. Lin, Z. H. Nie, and H. W. Ma, "Structural Damage Detection with Automatic Feature-Extraction through Deep Learning," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 12, pp. 1025–1046, Dec. 2017, doi: 10.1111/mice.12313.

[18] M. Kanojia, P. Kamani, G. S. Kashyap, S. Naz, S. Wazir, and A. Chauhan, "Alternative Agriculture Land-Use Transformation Pathways by Partial-Equilibrium Agricultural Sector Model: A Mathematical Approach," Aug. 2023, Accessed: Sep. 16, 2023. [Online]. Available: https://arxiv.org/abs/2308.11632v1

[19] G. S. Kashyap *et al.*, "Revolutionizing Agriculture: A Comprehensive Review of Artificial Intelligence Techniques in Farming," Feb. 2024, doi: 10.21203/RS.3.RS-3984385/V1.

[20] N. Marwah, V. K. Singh, G. S. Kashyap, and S. Wazir, "An analysis of the robustness of UAV agriculture field coverage using multi-agent reinforcement learning," *International Journal of Information Technology (Singapore)*, vol. 15, no. 4, pp. 2317–2327, May 2023, doi: 10.1007/s41870-023-01264-0.

[21] G. S. Kashyap, A. Siddiqui, R. Siddiqui, K. Malik, S. Wazir, and A. E. I. Brownlee, "Prediction of Suicidal Risk Using Machine Learning Models." Dec. 25, 2021. Accessed: Feb. 04, 2024. [Online]. Available: https://papers.ssrn.com/abstract=4709789

[22] G. S. Kashyap, K. Malik, S. Wazir, and R. Khan, "Using Machine Learning to Quantify the Multimedia Risk Due to Fuzzing," *Multimedia Tools and Applications*, vol. 81, no. 25, pp. 36685–36698, Oct. 2022, doi: 10.1007/s11042-021-11558-9.

[23] S. Wazir, G. S. Kashyap, and P. Saxena, "MLOps: A Review," Aug. 2023, Accessed: Sep. 16, 2023. [Online]. Available: https://arxiv.org/abs/2308.10908v1

[24] P. Kaur, G. S. Kashyap, A. Kumar, M. T. Nafis, S. Kumar, and V. Shokeen, "From Text to Transformation: A Comprehensive Review of Large Language Models' Versatility," Feb. 2024, Accessed: Mar. 21, 2024. [Online]. Available: https://arxiv.org/abs/2402.16142v1

[25] S. Naz and G. S. Kashyap, "Enhancing the predictive capability of a mathematical model for pseudomonas aeruginosa through artificial neural networks," *International Journal of Information Technology 2024*, pp. 1–10, Feb. 2024, doi: 10.1007/S41870-023-01721-W.

[26] G. S. Kashyap, A. E. I. Brownlee, O. C. Phukan, K. Malik, and S. Wazir, "Roulette-Wheel Selection-Based PSO Algorithm for Solving the Vehicle Routing Problem with Time Windows," Jun. 2023, Accessed: Jul. 04, 2023. [Online]. Available: https://arxiv.org/abs/2306.02308v1

[27] G. S. Kashyap, D. Mahajan, O. C. Phukan, A. Kumar, A. E. I. Brownlee, and J. Gao, "From Simulations to Reality: Enhancing Multi-Robot Exploration for Urban Search and Rescue," Nov. 2023, Accessed: Dec. 03, 2023. [Online].

Available: https://arxiv.org/abs/2311.16958v1

[28] H. Habib, G. S. Kashyap, N. Tabassum, and T. Nafis, "Stock Price Prediction Using Artificial Intelligence Based on LSTM– Deep Learning Model," in *Artificial Intelligence & Blockchain in Cyber Physical Systems: Technologies & Applications*, CRC Press, 2023, pp. 93–99. doi: 10.1201/9781003190301-6.

[29] G. S. Kashyap *et al.*, "Detection of a facemask in real-time using deep learning methods: Prevention of Covid 19," Jan. 2024, Accessed: Feb. 04, 2024. [Online]. Available: https://arxiv.org/abs/2401.15675v1

[30] S. Wazir, G. S. Kashyap, K. Malik, and A. E. I. Brownlee, "Predicting the Infection Level of COVID-19 Virus Using Normal Distribution-Based Approximation Model and PSO," Springer, Cham, 2023, pp. 75–91. doi: 10.1007/978-3-031-33183-1_5.

[31] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553. Nature Publishing Group, pp. 436–444, May 27, 2015. doi: 10.1038/nature14539.

[32] Chandratreya, Abhijit, et al. "Robotics and Cobotics: A Comprehensive Review of Technological Advancements, Applications, and Collaborative Robotics in Industry." International Journal of Intelligent Systems and Applications in Engineering, vol. 12, no. 21s, 22 Mar. 2024, pp. 1027–1039, ijisae.org/index.php/IJISAE/article/view/5501

[33] Kamuni, N., Jindal, M., Soni, A., Mallreddy, S. R., & Macha, S. C. (2024). A Novel Audio Representation for Music Genre Identification in MIR (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2404.01058

[34] Kamuni, N. ., A. Cruz, I. G. ., Jaipalreddy, Y., Kumar, R. ., & Pandey, V. K. . (2024). Fuzzy Intrusion Detection Method and Zero-Knowledge Authentication for Internet of Things Networks. International Journal of Intelligent Systems and Applications in Engineering, 12(16s), 289–296. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/4821

[35] Kumar, A., Dodda, S., Kamuni, N., & Vuppalapati, V. S. M. (2024). The Emotional Impact of Game Duration: A Framework for Understanding Player Emotions in Extended Gameplay Sessions (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2404.00526

[36] Dodda, S., Kumar, A., Kamuni, N., & Ayyalasomayajula, M. M. T. (2024). Exploring Strategies for Privacy-Preserving Machine Learning in Distributed Environments. Institute of Electrical and Electronics Engineers (IEEE). https://doi.org/10.36227/techrxiv.171340711.17793838/v1

[37] Dodda, Suresh, et al. "Exploring AI-Driven Innovations in Image Communication Systems for Enhanced Medical Imaging Applications." Journal of Electrical Systems, vol. 20, no. 3s, 4 Apr. 2024, pp. 949–959, journal.esrgroups.org/jes/article/view/1409/, https://doi.org/10.52783/jes.1409. Accessed 30 Apr. 2024

[38] H. Shah and N. Kamuni, "DesignSystemsJS - Building a Design Systems API for aiding standardization and AI integration," 2023 International Conference on Computing, Networking, Telecommunications & Engineering Sciences Applications (CoNTESA), Zagreb, Croatia, 2023, pp. 83-89, doi: 10.1109/CoNTESA61248.2023.10384889

[39] Kumar, A., Dodda, S., Kamuni, N., & Arora, R. K. (2024). Unveiling the Impact of Macroeconomic Policies: A Double Machine Learning Approach to Analyzing Interest Rate Effects on Financial Markets (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2404.07225

[40] Narne, Suman, et al. "AI-Driven Decision Support Systems in Management: Enhancing Strategic Planning and Execution." International Journal on Recent and Innovation Trends in Computing and Communication, vol. 12, no. 1, 16 Mar. 2024, pp. 268–276, www.ijritcc.org/index.php/ijritcc/article/view/10252. Accessed 6 May 2024

[41] Dodda, S., Kumar, A., Kamuni, N., & Ayyalasomayajula, M. M. T. (2024). Exploring Strategies for Privacy-Preserving Machine Learning in Distributed Environments. Institute of Electrical and Electronics Engineers (IEEE). https://doi.org/10.36227/techrxiv.171340711.17793838/v1

[42] Dodda, Suresh. "Suresh Dodda: Discussing Automated Payroll Process." Ceoweekly.com, 16 Apr. 2024, ceoweekly.com/discussing-automated-payroll-process-with-suresh-dodda/. Accessed 6 May 2024