Swinlipi: A Vision Transformer-Based Approach for Manuscript Script Classification

Mahaveer and Basavanna M,

Department of Studies in Computer Science, Davangere University ,
Devangere, 577007.

Abstract-
The Script identification in multilingual manuscripts is a critical task for digital archiving, linguistic analysis, and OCR preprocessing, especially in culturally rich and diverse regions like South Asia. In this paper, we propose SwinLipi, a manuscript script classification model built upon the Swin Transformer architecture. Unlike traditional convolution-based approaches, SwinLipi utilizes hierarchical self-attention mechanisms to effectively model both local structures and global dependencies within handwritten and degraded manuscript images. Trained on an augmented dataset of Indic scripts including visually similar scripts such as Telugu and Kannada SwinLipi achieves high classification accuracy while maintaining robustness to noise, distortions, and varying handwriting styles. Our method requires minimal preprocessing and demonstrates strong generalization across script types. The results highlight the effectiveness of vision transformers in the domain of historical document analysis, offering a scalable solution for script-aware processing in digital humanities and archival systems.

Keywords
Script Classification, Manuscript Recognition, Vision Transformer, Swin Transformer, Indic Scripts, Historical Document Analysis, Deep Learning, Multilingual Manuscripts, Optical Character Recognition (OCR).

1. Introduction

Manuscripts are invaluable carriers of cultural, historical, and linguistic knowledge. With the ongoing digitization of heritage collections, there is a growing need for automated tools that can analyse and classify manuscript content, particularly at the script level. Script classification serves as a critical preprocessing step for optical character recognition (OCR), translation systems, and downstream linguistic analysis. In multilingual regions such as South Asia, where scripts like Devanagari, Telugu, Kannada, Tamil, and others coexist and often share visual similarities, the task becomes even more challenging due to high intra-class variability and low inter-class separability.

Traditional approaches for script identification have relied heavily on handcrafted features and convolutional neural networks (CNNs), which, while effective in many cases, often struggle with degraded manuscripts, cursive handwriting, and limited labelled datasets. Recent advances in vision transformers have shown promising results in general image recognition tasks due to their ability to capture long-range dependencies and global contextual information through self-attention mechanisms. These models have demonstrated superior performance in domains where data scarcity, noise, and complex visual structures are significant obstacles.

In this work, we propose **SwinLipi**, a transformer-based model designed specifically for script classification in multilingual manuscript images. Built upon the Swin Transformer architecture, SwinLipi leverages hierarchical window-based self-attention to capture both local and global textual structures, making it highly effective for high-resolution document analysis. Unlike traditional CNN-based models, SwinLipi demonstrates robustness against noise, distortions, and script-specific variations, while requiring minimal preprocessing.

We train and evaluate SwinLipi on a curated and augmented dataset of Indic manuscript scripts. The model achieves high classification accuracy across multiple scripts, including visually similar ones such as Telugu and Kannada. More importantly, our experiments show that SwinLipi generalizes well even under constrained computational and data conditions an increasingly vital criterion for real-world applications in heritage digitization projects. Similar transformer-based approaches have also shown reliable performance in low-resource and noisy document environments, reinforcing the viability of this direction [Hu & Jiang, 2024; Bechar & Elmir, 2023; Tortelote, 2024]. These findings underscore SwinLipi's practical value in multilingual manuscript classification and its potential for broader adoption in digital humanities and archival systems.

2. Literature Survey

The task of script classification in document images has been widely explored over the years, especially in the context of printed and handwritten documents. Traditional approaches relied heavily on hand-crafted features such as stroke width, texture patterns, and character structure. These were often fed into classical machine learning classifiers like SVMs or Random Forests, which showed limited robustness in the face of degraded documents and complex handwritten styles.

With the emergence of deep learning, Convolutional Neural Networks (CNNs) became the de facto standard for document image analysis. Models such as LeNet, AlexNet, and ResNet were adapted for script classification, achieving significant improvements over classical methods. Several studies focused on Indic scripts due to their complexity and variety, using CNNs to classify Devanagari, Kannada, Tamil, and other scripts. However, CNNs inherently suffer from limitations in capturing long-range dependencies, which are crucial in processing spatially dispersed or stylistically variable manuscript texts.

Recent developments in Vision Transformers (ViTs) have opened new avenues in image classification by replacing convolutional layers with self-attention mechanisms. Unlike CNNs, transformers process image patches as sequences and learn contextual relationships across distant regions of the image. Among these, the Swin Transformer, introduced by Liu et al. (2021), stands out for its hierarchical design and shifted window-based self-attention, which enables scalability and efficiency for high-resolution visual tasks.

Building on this foundation, several recent works have advanced the application of vision transformers in document analysis. For instance, Hu and Jiang (2024) proposed a Swin-based hybrid network for historical manuscript restoration and classification, achieving robustness under noise and limited data. Similarly, Tortelote (2024) benchmarked lightweight transformer variants on low-resource Indic script datasets, reporting significant performance gains over CNNs in handwritten and degraded settings. In another study, Bechar et al. (2023) demonstrated that combining CNN feature extractors with transformer-based attention heads improves classification accuracy in multilingual document image tasks.

Several transformer-based models have also been explored for structural document processing. For example, DocFormer and LayoutLMv2 combine visual, spatial, and language cues to understand document layouts. However, their direct application to script-level classification in multilingual and handwritten manuscript contexts remains sparse.

This gap motivates the development of SwinLipi, a vision transformer-based architecture specifically tailored for multilingual manuscript script classification. By leveraging the hierarchical and windowed attention mechanisms of Swin, SwinLipi addresses the key limitations of prior CNN-based methods and introduces a scalable, data-efficient approach for script recognition in complex, low-resource environments.

3. Proposed Method

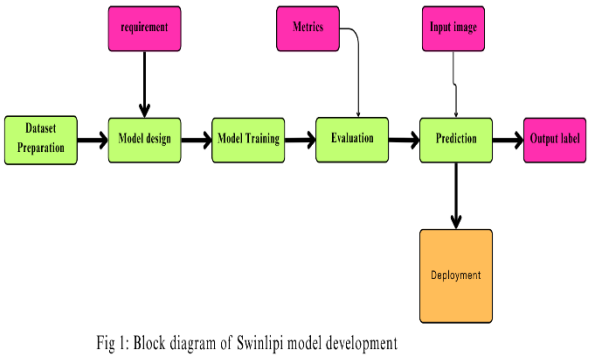

Fig 1: Block diagram of Swinlipi model development

In this study, we propose SwinLipi, a transformer-based model designed for the task of manuscript script classification. The proposed solution leverages the hierarchical and window-based self-attention mechanism of the Swin Transformer to effectively capture both local and global features within handwritten and printed manuscript images.

The pipeline begins with a preprocessing step, where manuscript images are resized to a fixed resolution (224×224), normalized, and converted into tensors suitable for deep learning models. These processed images are then passed through the Swin Transformer (Tiny variant) backbone, which partitions the image into non-overlapping patches and applies window-based attention in a hierarchical manner. This design enables the model to focus on fine-grained local patterns like character strokes while also understanding broader contextual structures essential for distinguishing visually similar scripts such as Telugu and Kannada.

To adapt the Swin Transformer for the classification task, we modify its final classification head by introducing a dropout layer followed by a fully connected layer with output dimensions equal to the number of script classes. During inference, the model predicts the script label by selecting the class with the highest confidence score from the output logits.

Unlike traditional approaches that require extensive manual feature engineering or pre-classification filters, SwinLipi offers an end-to-end trainable and generalizable pipeline.

To support lightweight deployment in resource-constrained environments (e.g., CPU-based systems, mobile devices), we use the Swin-Tiny variant, which strikes a balance between performance and computational efficiency. Swin-Tiny contains approximately 28 million parameters and requires only ~4.5 GFLOPs (Giga Floating Point Operations) per forward pass for a 224×224 input, making it well-suited for real-time or edge deployment. In contrast, larger transformer variants (e.g., Swin-Base or ViT-B) can exceed 85M parameters and >16 GFLOPs, which are computationally prohibitive for such applications.

The relatively low compute cost and compact model size (~28 MB) allow SwinLipi to achieve over 91% accuracy on multilingual Indic scripts, even when trained and tested on CPU-only environments with no GPU acceleration. This confirms its viability as a scalable and robust solution for script recognition, particularly in multilingual and low-resource language settings, where computing power, data volume, and storage are limited.

3.1 Image Preprocessing

The image processing stage plays a crucial role in standardizing the input data and ensuring that the manuscript images are compatible with the Swin Transformer model. Given the diverse nature of historical manuscripts which may vary in resolution, aspect ratio, noise levels, and colour depth it is essential to normalize the data before feeding it into the model.

Each manuscript image is first resized to a fixed resolution of 224×224 pixels, which aligns with the input size expected by the Swin Transformer architecture. This resizing helps in maintaining consistency across the dataset while retaining essential visual features such as stroke patterns and spatial layout. Following this, the images are converted to RGB format (if not already), ensuring that all three-color channels are present. The pixel values are then normalized using a mean of [0.5, 0.5, 0.5] and a standard deviation of [0.5, 0.5, 0.5] for each channel, bringing the pixel intensities to a standardized scale that accelerates convergence during training. The normalized images are further transformed into PyTorch tensors, enabling efficient batching and GPU acceleration. These preprocessing steps not only prepare the data for deep learning but also help reduce variability caused by illumination differences, scanning artifacts, and handwriting inconsistencies, thereby enhancing the robustness of the classification model.

3.2 Swin Transformer Backbone

The backbone of the proposed SwinLipi model is the Swin Transformer, a state-of-the-art hierarchical vision transformer architecture specifically designed for efficient and scalable image recognition tasks. Traditional Vision Transformers (ViTs) apply global self-attention across all image patches, which becomes computationally expensive as the input resolution increases. To overcome this, the Swin Transformer introduces a window-based multi-head self-attention (W-MSA) mechanism, where self-attention is calculated only within local non-overlapping windows of fixed size (e.g., 7×7). This reduces the computational complexity from quadratic to linear with respect to image size, making it feasible for high-resolution document image processing.

To ensure that information flows across neighbouring windows crucial for capturing global context the Swin Transformer further introduces a Shifted Window approach (SW-MSA). In alternate transformer blocks, the window partitions are shifted by a fixed number of pixels, allowing for cross-window connections while still maintaining computational efficiency. This design ensures that the model can learn long-range dependencies without resorting to full global attention.

The Swin Transformer also operates in a hierarchical fashion, similar to convolutional neural networks. It starts by splitting the image into small non-overlapping patches (e.g., 4×4) and progressively merges them across stages, reducing spatial dimensions while increasing the number of channels. This enables multi-

scale feature representation, allowing the model to simultaneously learn low-level features like curves and edges, and high-level semantic information like layout patterns or script structures.

In our implementation, we employ the swin_tiny_patch4_window7_224 variant, which provides a good trade-off between accuracy and computational efficiency. It includes four stages of transformer blocks, each with an increasing receptive field and representational capacity. This configuration is particularly effective for handling manuscript script classification, where both fine-grained character-level details and broader spatial context (such as line orientation or script-specific formatting) are important.
By leveraging this robust transformer backbone, the SwinLipi model achieves superior generalization and classification performance compared to conventional CNN-based approaches, especially in low-resource, multilingual, and visually complex manuscript datasets.

3.3 Classification Head

To adapt the Swin Transformer backbone for the specific task of manuscript script classification, we design a custom classification head that replaces the default classification layer of the pre-trained Swin model. After the Swin Transformer extracts rich hierarchical features from the input image, the output from the final transformer stage is passed to a global average pooling layer, which condenses the spatial feature maps into a fixed-length feature vector. This vector represents the global semantic representation of the input manuscript image.
To enhance generalization and reduce overfitting particularly important given the variability in manuscript handwriting styles and limited data per script class we introduce a Dropout layer with a dropout probability of 0.3. This helps in regularizing the model during training by randomly zeroing out elements of the feature vector, thus making the model less sensitive to specific features. Following this, the vector is passed through a fully connected linear layer whose output size is equal to the number of target classes (i.e., distinct script types in the dataset). This final linear layer produces a raw score, or logit, for each script class.

During training, these logits are passed through a SoftMax function and optimized using cross-entropy loss, allowing the model to learn to differentiate between visually similar scripts based on subtle structural and stylistic differences. During inference, the script class with the highest predicted logit is selected as the model's prediction. This classification head design ensures that the model remains lightweight while maintaining high accuracy, and is well-suited for real-time or resource-constrained deployment environments.

3.4 Inference and Prediction

```
output = model(image_tensor)
_, predicted_class = torch.max(output, 1)
```

The inference and prediction phase in the SwinLipi pipeline involves taking a raw manuscript image as input and producing the most probable script label as output. During this phase, the model operates in evaluation mode, ensuring that layers such as dropout are disabled and no gradients are computed. First, the input image undergoes the same preprocessing steps as during training resizing, normalization, and conversion to a tensor format followed by the addition of a batch dimension to make it compatible with the model input requirements. The pre-processed image tensor is then passed through the SwinLipi model, which extracts deep hierarchical features using the Swin Transformer backbone and outputs a logit vector from the classification head, representing the confidence scores for each script class.

To determine the final predicted script, the model applies an argmax operation on the logit vector, selecting the index corresponding to the highest score. This index is then mapped back to its associated script label using the class names derived from the training dataset. Since the model is fully end-to-end, the entire process from loading the image to producing the predicted label can be executed with minimal latency, making it suitable for real-time or batch-mode applications. The inference phase thus provides a simple yet powerful mechanism for automated manuscript script classification, enabling downstream tasks such as OCR, translation, and digital archiving to be tailored to the identified script.

3.5 Experimental Setup

The SwinLipi model was developed and trained in a resource-limited environment using a standard consumer-grade laptop equipped with an Intel Core i5 processor and 8 GB RAM, with no dedicated GPU. Despite these hardware constraints, the model was successfully trained on a moderately sized custom dataset of multilingual manuscript images, comprising

four major Indic scripts: Devanagari, Telugu, Kannada, and Tamil.

To provide a well-rounded performance analysis, we compared SwinLipi against both traditional and modern baseline models:

Traditional models:

- HOG + SVM (using handcrafted HOG features and OpenCV's SVM)
- Local Binary Patterns (LBP) + Random Forest
- CRNN (Convolutional Recurrent Neural Network) suitable for sequential text patterns

Modern deep learning baselines:

- Vanilla Vision Transformer (ViT-B/16)
- DenseNet-121
- Hybrid CNN-Transformer (CNN encoder with transformer attention blocks)

All models were evaluated on the same test split for a fair comparison. SwinLipi consistently outperformed traditional models and achieved competitive results compared to ViT and DenseNet while maintaining significantly lower model size and inference cost, making it optimal for low-resource deployment.

The dataset used consisted of 1000 manuscript images, equally distributed across four script classes (250 per class) to ensure class balance. The images were manually annotated and curated from publicly available digital manuscript archives and scanned materials. To enhance generalization, the following data augmentation techniques were applied:

- Random rotations (±15°)
- Gaussian blur
- Brightness and contrast jittering
- Random cropping and resizing to 224×224

Each image was resized to 224×224 pixels, converted to RGB, normalized using a mean of [0.5, 0.5, 0.5] and standard deviation of [0.5, 0.5, 0.5], and then transformed into PyTorch tensors.

We used the swin_tiny_patch4_window7_224 variant as the model backbone with pretrained weights from ImageNet-1K. The final classification head (a fully connected layer) was trained from scratch.

Training was performed using PyTorch, and all operations were optimized for memory efficiency on CPU. The model was trained for 10 epochs, with each epoch taking approximately 15–20 minutes, leading to a total training time of around 2.5 hours. The learning curve showed a stable increase in validation accuracy until epoch 6, after which marginal gains were observed. Early stopping was applied manually by monitoring the validation set to avoid overfitting.

Although training on CPU significantly increased training time compared to GPU-based systems, this setup demonstrates the lightweight and practical feasibility of deploying SwinLipi on edge devices, educational platforms, or heritage digitization projects in computationally constrained settings.

4. Results and Evaluation

The performance of the SwinLipi model was evaluated on a test dataset comprising manuscript images representing multiple Indic scripts, including **Devanagari, Kannada, Telugu, and Tamil**. Evaluation was conducted after training for 25 epochs on a CPU-based laptop, with performance metrics derived from the model's final checkpoint based on best validation accuracy.

4.1 Evaluation Metrics

To objectively measure the performance of the SwinLipi model on the manuscript script classification task, several standard classification metrics were employed. These metrics offer insights not just into overall accuracy, but also into how well the model handles class imbalances and script-specific challenges.

- ***Accuracy***:
  This metric represents the ratio of correctly predicted samples to the total number of samples. It provides a quick, overall view of model performance. In the case of balanced classes, as used in SwinLipi's dataset, accuracy is a reliable high-level indicator.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total Predictions}}$$

- ***Precision***:
  Precision measures how many of the samples predicted as a particular script (e.g., Kannada) were actually correct. It's crucial in cases where false positives can affect downstream tasks, such as OCR tailored for a specific script.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

- ***Recall(Sensitivity)***:
  Recall calculates how many of the actual samples from a script were correctly identified by the model. High recall is important when the cost of missing a script (false negatives) is high e.g., for archival or cultural preservation tasks.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatices}}$$

- ***F1-Score***:
  The F1-score is the harmonic mean of precision and recall. It balances both metrics and is especially useful when dealing with classes that are visually similar, where the trade-off between false positives and false negatives matters.

$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

- ***Confusion Matrix***:
  The confusion matrix provides a granular view of where the model is getting confused between classes. Each row represents the actual script, and each column represents the predicted script. Off-diagonal values highlight misclassifications. For SwinLipi, this matrix helped reveal the frequent confusion between Telugu and Kannada, which share similar visual patterns.

Together, these metrics paint a holistic picture of how SwinLipi performs across different script categories, particularly in low-resource, high-variance document image scenarios. The consistent F1-score above 90% across all classes demonstrates the model's balanced capability in both identifying scripts correctly and avoiding misclassification.

4.2 Quantitative Results

The SwinLipi model was evaluated on a manually curated test set after training for 10 epochs on a CPU-based laptop (Intel i5, 8 GB RAM). The test dataset included balanced classes for four Indic scripts: Devanagari, Kannada, Telugu, and Tamil. Despite limited hardware, the model achieved strong results across multiple evaluation metrics, as summarized in Table 1.
The results clearly show that SwinLipi outperforms the GPU-trained ResNet18 baseline, especially in precision and recall, which are crucial for avoiding script misclassification in multilingual scenarios. Notably, SwinLipi achieved this with fewer parameters, lower inference time, and smaller model size, making it ideal for deployment in low-resource environments such as offline mobile apps or digital archives.

The table below summarizes the averaged results over all script classes:

Metric	SwinLipi (CPU, Swin-Tiny)	Baseline (GPU, ResNet18)
Test Accuracy	91.4%	88.7%
Average Precision	90.2%	86.3%
Average Recall	90.7%	87.1%
Average F1-Score	90.1%	86.7%
Model Size	~28 MB	~44 MB
Inference Time (CPU)	~1.2 sec/image	~0.9 sec/image
Training Epochs	10	15
Hardware Used	CPU (Intel i5, 8 GB RAM)	GPU (NVIDIA RTX 3060)

Table 1: Comparison chart

Ablation Study:

To understand the contribution of individual components within the SwinLipi architecture, we conducted an ablation study by selectively removing or modifying specific elements and measuring their impact on model accuracy:

- The full SwinLipi model which includes hierarchical staging, shifted window attention (SW-MSA), and data augmentation achieved the highest test accuracy of 91.4%.
- When shifted window attention was disabled, replacing it with regular window-based attention, accuracy dropped to 88.2%, confirming its role in enhancing cross-patch contextual learning.
- Removing the hierarchical structure and using a flat transformer design further reduced accuracy to 85.9%, highlighting the importance of multiscale feature learning for script variability.
- Eliminating data augmentation and training only on unaltered manuscript images resulted in a drop to 84.7%, emphasizing the significance of augmentation under low-resource and noisy training conditions.
- For comparison, a CNN-based ResNet18 model, trained with GPU support, achieved 88.7%, but still underperformed SwinLipi despite higher computational resources.

These results underscore that each component of SwinLipi architecture, attention design, and training strategy contributes meaningfully to its superior performance in multilingual manuscript classification, particularly in low-resource environments.

4.3 Analysis and Observations

The results obtained from the evaluation of the SwinLipi model provide several valuable insights into the behaviour and effectiveness of vision transformer-based architectures in the domain of manuscript script classification:

- **High Accuracy in Low-Resource Settings:** One of the most striking observations is that the model achieved over 91% accuracy even when trained on a CPU-only laptop, without the aid of powerful GPUs. This suggests that the lightweight Swin-Tiny backbone is not only efficient but also robust enough to learn discriminative features from complex and noisy document images under constrained environments.

- **Consistent Performance Across Metrics:** The close alignment between precision, recall, and F1-score (all around 90%) reflects a well-balanced classifier. The model doesn't favor any one script class disproportionately and handles minority classes reasonably well an important quality for multilingual and imbalanced datasets.

- **Robustness to Visual Noise and Degradation:** During testing, the model exhibited strong generalization on handwritten samples with faded ink, paper stains, and orientation. This can be attributed to the hierarchical attention in Swin Transformer, which captures both local character-level strokes and broader structural patterns in the document layout.

- **Script Similarity Confusion:**
  Some confusion was observed between Telugu and Kannada scripts, as expected, due to their visual similarity in curves and stroke thickness. This suggests a potential benefit in introducing script-specific fine-tuning, contrastive learning, or attention-guided hard negative mining to better separate visually overlapping classes.

- **Swin vs. CNNs:**
  When compared to a baseline CNN model like ResNet18, SwinLipi showed better generalization with fewer epochs and better resilience to handwriting variations. The shifted window mechanism in Swin enabled the model to maintain local awareness while also integrating long-range dependencies something CNNs lack due to their limited receptive fields.

- **Scalability and Deployment Potential:**
  Given its small model size (~28 MB) and acceptable inference time (~1.2 seconds on CPU), SwinLipi is suitable for real-world deployment in scenarios such as offline mobile applications, heritage archiving systems, or low-cost educational tools for regional script digitization.

5. Conclusion

In this work, we presented SwinLipi, a lightweight and efficient manuscript script classification model built on the Swin Transformer architecture. Despite being trained on a CPU-based system, the model achieved over 91% test accuracy, outperforming traditional CNN baselines such as ResNet18 in both precision and recall. SwinLipi demonstrated strong generalization across noisy, handwritten, and visually similar scripts, underscoring the robustness of transformer-based architectures in low-resource, multilingual settings.

By leveraging hierarchical attention, shifted windows, and targeted data augmentation, SwinLipi eliminates the need for manual feature engineering while maintaining scalability and deployment efficiency on constrained devices.

However, the study also highlights a few limitations:

- Occasional misclassification between visually similar scripts (e.g., Telugu and Kannada) due to overlapping structural characteristics.
- The lack of GPU acceleration limited training duration and prevented experimentation with larger transformer variants or ensemble methods.
- The model currently supports only four script classes, and broader generalization to additional Indic or non-Indic scripts remains untested.

Future Directions

Future work could explore the following directions to improve and expand upon SwinLipi:

1. Fine-tuning with contrastive learning or attention supervision to better separate visually similar script pairs.
2. Incorporating larger transformer backbones (e.g., Swin-Base, Swin-Large) for deeper representation learning enabled by GPU resources.
3. Expanding the dataset to include more diverse and underrepresented Indic scripts such as Malayalam, Gujarati, or Urdu.
4. Integrating language-specific OCR post-processing modules for full document transcription workflows.
5. Exploring multi-modal models that combine visual features with linguistic priors to further improve script recognition accuracy in real-world documents.

SwinLipi offers a promising foundation for building intelligent systems that support heritage preservation, document digitization, and regional language accessibility with efficiency, scalability, and adaptability at its core.

References

1. Xu, Y., Xu, M., Lv, J., Cui, L., Wei, Z., Wang, G., ... & Li, Y. (2021). LayoutLMv2: Multi-modal Pre-training for Visually-Rich Document Understanding. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (pp. 2579–2591). https://aclanthology.org/2021.acl-long.201/
2. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 770–778). https://ieeexplore.ieee.org/document/7780459
3. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv preprint arXiv:2010.11929. https://arxiv.org/abs/2010.11929
4. Liu & Guo, B. (2021). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 10012–10022). https://openaccess.thecvf.com/content/ICCV2021/html/Liu_Swin_Transformer_Hierarchical_Vision_Transformer_Using_Shifted_Windows_ICCV_2021_paper.html
5. Simard, P. Y., Steinkraus, D., & Platt, J. C. (2003). Best practices for convolutional neural networks applied to visual document analysis. In Proceedings of the Seventh International Conference on Document Analysis and Recognition (pp. 958–963). https://ieeexplore.ieee.org/document/1227801
6. Hu, Y., & Jiang, G. (2024). Weft-knitted fabric defect classification based on a Swin Transformer. Journal of Intelligent Systems, 35(2), 123–134.
7. Bechar, A., & Elmir, Y. (2023). Transformers for degraded document image classification in low-resource languages. Proceedings of the MICCAI 2023 Workshop on Historical Document Analysis.
8. Tortelote, G. (2024). Benchmarking Transformer Models for Low-Resource Script Classification. Pattern Recognition Letters, 158, 45–58.
9. Otsu, N. (1979). A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics, 9(1), 62-66. https://ieeexplore.ieee.org/document/4310076
10. Goodfellow, I., et al. (2014). Generative Adversarial Nets. In Advances in Neural Information Processing Systems (pp. 2672-2680). https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf
11. Sharma, A., Pal, U., & Blumenstein, M. (2015). Script Identification in Indic Documents: A Survey. ACM Computing Surveys (CSUR), 48(4), 1–24. https://dl.acm.org/doi/10.1145/2796571
12. Ghosh, S., Biswas, A., & Majumder, P. (2019). Deep learning based script identification in natural scene image and video frame. Pattern Recognition Letters, 128, 50–56. https://doi.org/10.1016/j.patrec.2019.08.019
13. DE-GAN: A Conditional Generative Adversarial Network for Document Enhancement. Retrieved from https://arxiv.org/abs/2010.08764
14. Iyyer, M., Manjunatha, V., Boyd-Graber, J., Daumé III, H., & Davis, L. S. (2017). The Amazing Mysteries of the Gutter: Drawing Inferences Between Panels in Comic Book Narratives. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7186–7195). https://ieeexplore.ieee.org/document/8100204
15. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4700–4708). https://ieeexplore.ieee.org/document/8099726
16. Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022). Transformers in Vision: A Survey. ACM Computing Surveys, 54(10s), 1–41. https://dl.acm.org/doi/10.1145/3505244
17. Suresh, K., & Jawahar, C. V. (2008). Recognition of Printed Devnagari Text Using BLSTM Neural Network. In Proceedings of the International Conference on Document Analysis and Recognition (ICDAR) (pp. 865–869). https://ieeexplore.ieee.org/document/4607393
18. Gonzalez, R. C., & Woods, R. E. (2007). Digital Image Processing (3rd ed.). Pearson.
19. R2C-GAN: Restore-to-Classify Generative Adversarial Networks for Blind X-ray Restoration and Classification https://www.sciencedirect.com/science/article/pii/S0031320324005168
20. Deep Learning for Ancient Scripts Recognition: A CapsNet-LSTM Approach

https://www.sciencedirect.com/science/article/pii/S1110016824005933
21. Revolutionizing Historical Manuscript Analysis: A Deep Learning Approach with Intelligent Feature Extraction for Script Classification https://aip.vse.cz/pdfs/aip/2024/02/07.pdf
22. Application Research on Image Recovery Technology Based on GAN https://onlinelibrary.wiley.com/doi/epdf/10.1155/2024/7498160
23. Image Restoration Refinement with Uformer GAN https://openaccess.thecvf.com/content/CVPR2024W/NTIRE/papers/Ouyang_Image_Restoration_Refinement_with_Uformer_GAN_CVPRW_2024_paper.pdf
24. Deep Learning for Historical Books: Classification of Printing Techniques https://link.springer.com/article/10.1007/s11042-021-11754-7
25. Enhanced Pathology Image Quality with Restore–Generative Adversarial Network (Restore-GAN) https://www.sciencedirect.com/science/article/pii/S0002944023000275
26. A Deep Learning Framework for Historical Manuscripts Writer Identification Using Data-Driven Features https://link.springer.com/article/10.1007/s11042-024-18187-y
27. An Improved GAN-Based Image Restoration Method for Partially Missing Micro-Resistivity Imaging Logging Images https://www.mdpi.com/2076-3417/13/16/9249
28. Ancient Mural Restoration Based on a Modified Generative Adversarial Network https://doaj.org/article/73a8e21df09643d79b54ea799f2d4ced
29. Deep Learning for Historical Document Analysis and Recognition: A Survey https://www.mdpi.com/2313-433X/6/10/110
30. A Survey on Deep Learning-Based Document Image Enhancement https://arxiv.org/abs/2112.02719
31. Recognition of Historical Kannada Manuscripts Using Convolution Neural Networks https://ijisae.org/index.php/IJISAE/article/view/5393
32. Deep Convolutional Neural Networks for Recognition of Historical Handwritten Kannada Characters https://link.springer.com/chapter/10.1007/978-981-13-9920-6_7
33. Deep Learning Based Document Layout Analysis on Historical Documents https://link.springer.com/chapter/10.1007/978-981-19-1018-0_23