SUMMARY
(PW24_VG03)

PAPER1: Recognition of Printed Mathematical Formula Symbols Based on Convolutional Neural Network

Dataset and Preprocessing

The dataset used for the recognition of printed mathematical formula symbols was self-created, as there are currently limited publicly available datasets for this task. It contains 6,300 images representing 134 categories of commonly used mathematical symbols. Before feeding the data into the network, the images underwent preprocessing steps such as binarization, filtering, refinement, and normalization to ensure consistent input.

Model

The model constructed is a Convolutional Neural Network (CNN) with five layers. It comprises two convolutional layers, two max-pooling layers, and a fully connected layer. The convolution layers use kernels of varying sizes (3x3, 5x5, and 9x9) to extract features, while the ReLU activation function and dropout techniques are applied to reduce overfitting and improve the generalization ability of the network. The CNN is specifically designed to recognize formula symbols by focusing on weight sharing and minimizing learning parameters.

Training

The model was trained using the stochastic gradient descent algorithm and implemented using the Caffe deep learning framework. The training set was divided into 168 batches, each containing 32 samples, to shorten convergence time. Multiple experiments were conducted to determine the optimal parameters, including kernel size and number, to maximize the recognition accuracy. Additionally, dropout was applied to the fully connected layer to prevent overfitting during training.

Results

The optimal configuration achieved a recognition accuracy of 94.25%. The model performed well for symbols with distinct shapes but struggled with symbols having subtle differences or distortions. Misclassification was observed primarily in symbols with similar shapes. The experiments showed that increasing the number of convolutional kernels and layers initially improved performance, but beyond a certain point, it led to diminishing returns.

Future Work

The primary limitation identified was the model's inability to distinguish symbols with minor shape variations and its lack of contextual understanding. Future work will focus on improving the recognition of such symbols by incorporating semantic analysis and context-aware approaches to enhance classification performance further.

PAPER-2: An Application for Automated Recognition and Processing of Handwritten Mathematical Equations.

Dataset and Preprocessing

The dataset used in this study is sourced from the publicly available Kaggle handwritten math symbols dataset. The original dataset contains 100,000 images of various math symbols, alphanumeric characters, and math operators, with each image sized at 45x45 pixels. To enhance recognition accuracy, a subset of the dataset containing 26 relevant classes (digits, arithmetic operators, equal signs, and variable symbols) was selected. Preprocessing steps such as denoising, RGB to grayscale conversion, and binarization were applied to the images to improvedata quality and remove redundant features like color information.

Model

The model implemented is a custom Convolutional Neural Network (CNN) designed for symbol recognition. The architecture consists of several layers:

1. Convolutional Layer with 32 filters of size (5x5).
2. ReLU Activation Layer.
3. MaxPooling Layer with a pool size of (2x2).
4. Dropout Layer with a drop probability of 0.2.
5. Two Fully Connected Layers—one with 128 neurons and the other with 26 neurons (corresponding to 26 symbol classes).
6. Softmax Activation Layer for output.

This setup was designed to identify basic mathematical symbols with high accuracy, leveraging a combination of convolutional and pooling layers to extract features and reduce dimensionality.

Training

The CNN model was trained on the subset of the Kaggle dataset using the Adam optimizer and categorical cross-entropy loss function for 14 epochs. The training set contained 110,011 samples, while the validation set included 12,236 samples. Data augmentation techniques like upsampling and downsampling were used to balance the class distributions within the dataset.

Results

The model achieved a cross-validation accuracy of 99.2% on the validation set. The high accuracy demonstrates the effectiveness of the custom CNN for handwritten symbol recognition. The application is capable of solving simultaneous equations, plotting graphs, and performing basic arithmetic computations based on image inputs or formatted text strings.

Future Work

The current application is limited by the image segmentation algorithm's ability to handle non-continuous symbols and noisy backgrounds. Enhancements such as using local binarization techniques and robust segmentation methods can improve the model’s performance on more complex images. Future work includes expanding the dataset to cover more mathematical symbols and different writing styles, as well as integrating support for solving systems of equations and differential equations. Additional improvements in the character recognition algorithm could further boost accuracy and make the application more versatile in diverse scenarios

Paper-3 : Recognition of Online Handwritten Mathematical Expressions Using Convolutional Neural Networks

Dataset

The dataset used in this paper comes from the CROHME competition, which provides training and testing data in Ink Markup Language (InkML) format. The dataset includes:

- Training set: 16,970 symbols and 2,259 expressions
- Testing set: 8,195 symbols and 836 expressions
- The dataset contains 75 different mathematical symbols. Each expression in InkML is represented as X-Y coordinates of pen-tip movements.

Model

The model built for this research relies on Convolutional Neural Networks (CNNs) for character-level classification, with the following architecture:

- CNN layers: Various architectures were tested, with configurations like 1–4 convolutional layers followed by max-pooling and fully connected layers.
- The best-performing CNN configuration consisted of 3 convolutional layers and 1 fully connected layer.
- The CNN was paired with Hidden Markov Models (HMMs) to leverage contextual information for expression-level classification.

Training

- Data Augmentation: The dataset was enriched by distorting images using an interpolation scheme to slightly displace pixel values, effectively doubling the size of the dataset.
- Neural Networks (NNs): Fully connected NNs were tested but did not perform as well as CNNs.
- CNN Tuning: Parameters like filter size, learning rate, regularization, momentum, and batch size were extensively tuned. The best CNN achieved 90% accuracy on character-level classification.
- HMM Integration: The probabilities from the CNN were used as inputs to the HMM, which improved expression-level accuracy by considering the sequence of symbols.

Results

- Character-Level Classification: The CNN achieved 90% accuracy, outperforming traditional SVM models (87%).
- Expression-Level Classification: Using the CNN in conjunction with HMMs achieved an accuracy of 41% with no symbols misclassified, improving over the baseline SVM.

- Segmentation Accuracy: The heuristic + CNN approach achieved 93% symbol-level accuracy and 56% expression-level accuracy, slightly outperforming the heuristic + SVM model.

Future Work

The authors identified several areas for improvement:

1. Segmentation Improvement: The current segmentation method has a significant impact on accuracy. They propose testing multiple segmentation hypotheses and including them in the HMM.
2. Spatial Information: Future work could incorporate spatial information to recognize complex mathematical symbols such as fractions or summations.
3. Size Information: Misclassification of symbols like commas suggests that incorporating size information into the model could improve performance.
4. Deeper CNNs: Training deeper CNNs with smaller filter sizes and additional data augmentation (e.g., rotations and affine transformations) could further boost accuracy.

PAPER-04 Multi-Feature_Learning_by_Joint_Training_for_Handw.pdf

Dataset: **CROHME** (Competition on Recognition of Online Handwritten Mathematical Expressions) datasets, which are widely used in the field of handwritten formula recognition

Model: The authors propose a new model called **SE-MCNN** (Squeeze-Extracted Multi-feature Convolutional Neural Network), which integrates both **online** and **offline** features. Key aspects of the model include:

Feature Extraction:

Online mode extracts eight-directional features to compensate for lost dynamic trajectory data. Offline mode uses Gabor filters for multi-directional gradient features.

SE-MCNN Framework:

Combines online and offline eight-directional features to capture stroke trajectories and spatial gradients. The network uses squeeze-extracted features and joint training to improve robustness and generalization in symbol classification

Result:

- The SE-MCNN model achieved **92.96%** top-1 accuracy on the **CROHME 2016** test set.
- The model also outperformed other state-of-the-art methods on the **CROHME 2014** test set, with a top-1 accuracy of **92.44%**.

Comparison to Other Models:

- **VGG-HMS**: 91.82% on CROHME 2014, 92.42% on CROHME 2016 (offline features)
- **MLP+RNN**: 91.04% on CROHME 2014, 92.81% on CROHME 2016 (online + offline features)
- **CNN+LSTM**: 91.28% on CROHME 2014, 92.27% on CROHME 2016 (online + offline features).

Paper-05

Dataset: The EMNIST dataset is used, which consists of handwritten characters in a 28x28 pixel format. It includes 88,800 training images and 14,800 test images.

Preprocessing: Techniques like image cropping, flipping, and hue adjustments are applied to enhance the dataset.

Model: The network includes two convolutional layers with max-pooling and fully connected layers followed by a dropout mechanism to prevent overfitting.

Training: The model is trained with a batch size of 256, and the dropout is applied at different layers to improve generalization

Results: The system achieved an accuracy of 90.2% after processing 145,600 images over 50 iterations, proving its effectiveness for recognizing handwritten English alphabets.

PAPER-06 Offline Handwritten Mathematical Expression Recognition using Convolutional Neural Network.

- **Dataset**: A dataset of 83 different mathematical symbols (numbers, Greek letters, operators) is compiled.
- **Preprocessing**: Techniques like binarization, noise reduction using median filtering, and thinning are applied to clean the images.
- **Segmentation**: The expressions are segmented into individual symbols using methods like recursive projection profiling and connected component labeling, preserving the order of the symbols.
- **Model:** SpNet-CNN, is used with three convolutional and three pooling layers to classify the symbols into 83 different classes. **ReLU** is used as the activation function, and **softmax** is employed in the final layer for classification.
- CNN is trained with 100 epochs and a learning rate of 1.0.

Results:

The proposed system is primarily focused on recognizing isolated symbols with a promise of future improvements for handling merged or connected symbols. It outputs the recognized symbols in LaTeX format.

Future work includes improving the segmentation for more complex, merged symbols.

PAPER-07 Recognition of Online Handwritten Math Symbols Using Deep Neural Networks.

- **Methods:** **Online Recognition**: Online methods deal with recognizing strokes as they are drawn (using time-sequenced data), traditional method for online is **Markov Random Fields (MRF).**
- **Offline recognition:** offline methods treat the input as a static image. Traditional method for offline is **Modified Quadratic Discriminant Function (MQDF).**
- **Dataset**: The experiments were conducted using the CROHME (Competition on Recognition of Online Handwritten Mathematical Expressions) database, which contains a large collection of mathematical symbols.
- **Model:** Convolutional Neural Networks (CNNs) and Bidirectional Long Short-Term Memory (BLSTM) networks, for the recognition of handwritten mathematical symbols.

Results:

Deep maxout convolutional network achieved **89.39%** on the **TestCROHME2014** dataset.

BLSTM Network achieved **87.29%** on the **TestCROHME2014** dataset.

When combined, DMCNs and BLSTM networks achieved an accuracy of **91.28%** (1-best rate) on the **TestCROHME2014** dataset.

The combined approach also achieved high top-3 and top-5 accuracies, with a **98.31%** top-3 rate and a **99.12%** top-5 rate on the same dataset.

PAPER-08 Handwritten Character Recognition Using Convolutional Neural Network

- **Dataset:** The researchers used the **NIST** dataset, which contains images of handwritten characters.
- **Model:** CNN architecture with convolutional layers, pooling layers, and fully connected layers.
- **Methodology:** **Pre-processing, Segmentation, Feature Extraction, Classification and Recognition**.
- **Results:**

The performance of the CNN model improved as the number of training images increased. The results are as follows:

With 200 training images, the accuracy was **65.32%**.
With 1000 training images, the accuracy reached **92.91%**.