Title: Date Extraction from Identification Documents using CRNN-Based OCR

1. Abstract

The purpose of this report is to detail the development of a Convolutional Recurrent Neural Network (CRNN) model for date extraction from identification documents. The project encompasses image preprocessing, network architecture selection, class implementation, and deployment in a Django-based web application. The CRNN-based OCR model was chosen for its effectiveness in handling sequential data, making it suitable for recognizing dates in diverse formats found in identification documents.

2. Introduction

Identification documents, such as driver's licenses, passports, and ID cards, often contain important information such as Date of Birth. Extracting this information accurately is crucial for various applications, including identity verification and document digitization. This project focuses on the development of a machine learning model (Supervised Learning Approach) capable of automatically extracting dates from images of identification documents.

3. Problem Description

The main problem addressed in this project is the extraction of date information from images of identification documents. To tackle this problem, a CRNN-based OCR model is proposed, as it has shown promise in handling sequential data and text recognition tasks.

4. Thought Process

4.1 Image Preprocessing Pipeline

To prepare images for the CRNN model, the following preprocessing steps were applied:

- **Image Loading**: Images were loaded using the OpenCV library.
- **Resizing**: Images were resized to a fixed size (250x40 pixels) to ensure consistency.
- **Grayscale Conversion**: Images were converted to grayscale to simplify processing.
- **Normalization**: Pixel values were normalized to the range [0, 1].
- **Dimension Expansion**: Image dimensions were expanded to match the CRNN model's input shape.

4.2 Network Architecture Choice

The choice of the CRNN architecture for date extraction was influenced by its successful application in Optical Character Recognition (OCR) tasks. The CRNN model combines Convolutional Neural Networks (CNNs) for feature extraction and Long Short-Term Memory (LSTM) networks for sequence modeling. This

architecture is suitable for recognizing dates in various formats, including handwritten and printed text.

4.2.1 Three Output Layers

In the chosen CRNN-based architecture for date extraction, three output layers were used, each responsible for predicting a different component of the date (day, month, and year). This architectural choice was made to address the specific nature of date extraction from identification documents, where dates are typically represented in a structured format (e.g., "dd/mm/yyyy" or "mm/dd/yyyy").

Importance of Three Output Layers:

1. **Structured Information Extraction:** Dates in identification documents are structured, consisting of day, month, and year components. By using three separate output layers, the model can learn to predict each component independently. This approach ensures that even if one component is written less clearly or varies in format, the model can still extract the other components accurately.

2. **Flexibility in Date Formats:** Different countries and regions may use varying date formats (e.g., "dd/mm/yyyy" in Europe and "mm/dd/yyyy" in the United States). By predicting each component separately, the model can adapt to these format differences without needing extensive preprocessing.

3. **Performance and Interpretability:** Using multiple output layers allows for fine-grained evaluation of the model's performance. We can assess the accuracy of day, month, and year predictions individually, providing insights into which components may require further improvement.

4.2.2 Choice of Loss Function and Performance Metric

The choice of loss function and performance metric plays a crucial role in training and evaluating the CRNN-based date extraction model.

Loss Function:

For each of the three output layers (day, month, and year prediction), the chosen loss function is **'sparse_categorical_crossentropy'**. Here's why this loss function was selected:

- **Categorical Crossentropy:** This loss function is well-suited for multiclass classification tasks like predicting day, month, and year components of a date. It measures the dissimilarity between predicted and true class probabilities and encourages the model to assign high probabilities to the correct class.

- **Sparse**: The 'sparse' variant of categorical crossentropy is used when labels are integers representing class indices (e.g., 0 for January, 1 for February). This is appropriate for our case since the date components are integer values.

Performance Metric:

The chosen performance metric is **'sparse_categorical_accuracy'** for each of the three output layers. This metric evaluates the accuracy of the model's predictions, taking into account the correct class indices. Here's why this metric is suitable:

- **Accuracy:** Accuracy is a fundamental metric for classification tasks. It measures the proportion of correctly predicted date components (day, month, or year) out of the total predictions. It provides an easily interpretable measure of model performance.

- **Sparse Variant:** The 'sparse' variant of categorical accuracy is used to match the use of 'sparse_categorical_crossentropy' as the loss function. This ensures consistency in how both predictions and labels are handled.

By using these loss functions and performance metrics for each output layer, the model is trained to minimize the dissimilarity between predicted and true date components while providing interpretable accuracy metrics for evaluation. This choice contributes to the overall effectiveness of the CRNN-based date extraction model.

4.2.3 Supporting Research

Several papers and projects have demonstrated the effectiveness of CRNN models for OCR tasks. Notable examples include "An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition" by Baoguang Shi et al. (2017). This paper introduced the CRNN architecture and showcased its capabilities in recognizing text sequences from images.

5. Class Implementation

The CRNN-based date extraction model was implemented as a Python class, **CRNNModel**, with the following key functionalities:

- Model construction and training.
- Image preprocessing methods.
- Label loading and encoding.
- Dataset loading and splitting.
- Prediction and evaluation functions.
- Model saving and loading.
- Conversion to ONNX and TensorRT formats for deployment optimizations.

6. Deployment

The CRNN-based OCR model was deployed using the Django web framework to create a user-friendly interface for date extraction from identification documents. The deployment process included the following steps:

- Integration with Django: The model was integrated into a Django web application.
- User Interface (UI): A user-friendly interface was created for users to upload identification documents.
- Predictive Service: The model provided real-time predictions on uploaded images.
- Result Display: Extracted date information was displayed to users.

7. Conclusion

The development of a CRNN-based OCR model for date extraction from identification documents addressed a real-world problem with practical applications. The chosen architecture, inspired by previous research in OCR, proved effective in recognizing date information from diverse document formats. The deployment as a web application further enhances its accessibility and usability.

This project highlights the importance of leveraging deep learning techniques and neural network architectures for solving complex data extraction tasks in various domains. The combination of image preprocessing, network architecture choice, and deployment in a user-friendly interface showcases technical skills and a holistic approach to solving practical problems.