OCR Arabic Birthdates

Egyptian ID Card Data

BLNK

Presented By:

Eng. Abdalrhman M.Alsisi

Social Links:

[Email]

[GitHub]

[LinkedIn]

Introduction

Welcome to the "OCR Arabic Birthdates" project. This endeavor aimed to create a model that can extract Arabic birthdates from images, treating numbers as characters. The project's dual objectives were to develop this model and deploy it using Django, showcasing the potential of OCR technology.

In this report, we'll explore the methods, dataset, steps to achieve a **100%** accuracy score, and the deployment process. Our journey will unveil the fusion of CNNs for image feature extraction and LSTM for text recognition, highlighting the power of machine learning and data science in solving real-world challenges.

Let's dive into the journey of converting visual content into actionable data.

Objectives

- Create a model capable of extracting Arabic birthdates from images.
- Deploy the model using Django, providing a web-based interface for date recognition.

Methodology

We will delve into the methodology employed to tackle the challenge of recognizing Arabic birthdates from images. The approach used a combination of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), specifically Long Short-Term Memory (LSTM) networks, along with the use of CTC (Connectionist Temporal Classification) loss.

1. Convolutional Neural Networks (CNNs)

   CNNs were employed for image feature extraction. This involved:

   - Preprocessing input images to enhance quality.
   - Designing a CNN architecture to capture relevant features.
   - Training the CNN on a dataset of Arabic birthdate images.

2. Recurrent Neural Networks (RNNs) - LSTM

   Following feature extraction, Long Short-Term Memory (LSTM) networks were used for textual recognition. This process included:

   - Transforming CNN's output into sequences.
   - Designing an LSTM network for sequence-to-sequence conversion.
   - Training the LSTM to recognize and decode Arabic date characters.

3. Connectionist Temporal Classification (CTC) Loss

   To handle variable-length sequences and align them with ground truth, the CTC loss function was employed. It played a critical role in training the model to accurately recognize Arabic dates.

4. Model Evaluation

   The model's performance was evaluated using accuracy, character-level accuracy, precision, recall, and F1-score. A validation dataset was used to assess its generalization ability.

By combining CNNs and LSTM networks, this methodology aimed to bridge the gap between visual content and textual recognition, enabling accurate extraction of Arabic birthdates from images.

Dataset

We will provide an overview of the dataset used for the "OCR Arabic Birthdates" project. A comprehensive dataset is the foundation of any machine learning endeavor, and this project was no exception.

1. Dataset Composition

   The dataset consisted of approximately 19,980 images, each containing an Arabic birthdate.

2. Ground Truth Text Files

   In conjunction with the images, there were corresponding text files, totaling another 19,980 entries. These text files contained the precise date information as shown in each image, serving as the ground truth for model training and evaluation.

Steps Taken

We will provide a step-by-step account of the actions taken during the "OCR Arabic Birthdates" project. These steps were instrumental in achieving a remarkable 100% accuracy score in recognizing Arabic birthdates from images.

1. Importing Libraries

   The project began by importing essential libraries and frameworks, including popular deep learning libraries like TensorFlow. These libraries provided the tools necessary for image processing and model creation.

2. Exploratory Data Analysis

   A crucial initial phase involved exploratory data analysis (EDA). This step aimed to gain insights into the dataset, understand its characteristics, and identify any potential challenges or patterns.

3. Data Preprocessing

   Data preprocessing played a pivotal role in preparing the dataset for model training. This phase included:

   - Developing function to convert characters to numerical values.
   - Creating a function to convert numerical values back to characters.
   - Implementing encoding function for each sample to facilitate training.

4. Preparing Train and Validation Datasets

   To ensure model robustness and assess its generalization, the dataset was split into training and validation subsets. Proper data shuffling and batching were employed.

5. Model Creation and Training

   The core of the project involved model creation and training. This encompassed designing a CNN-LSTM architecture, configuring loss function (CTC loss), and training the model on the prepared dataset.

6. Evaluation of the Model

   Model evaluation was conducted using accuracy. Validation data was used to assess the model's performance.

7. Testing on Real Images

   To validate real-world applicability, the model was tested on images containing Arabic birthdates. This step demonstrated the model's efficacy in practical scenarios.

8. Model Conversion to ONNX and TensorRT (TRT)

   To optimize the model for deployment, it was converted to ONNX format and further optimized using TensorRT (TRT), enhancing its inference speed and efficiency.

These meticulously executed steps collectively led to the successful development of a robust Arabic birthdate recognition model, achieving a 100% accuracy score.

Model Deployment

In this section, we discuss the deployment of the Arabic birthdate recognition model using Django. The deployment focused on creating an endpoint accessible through API calls, without a user interface. Key points include:

- Utilization of Django for deployment.
- Creation of an API endpoint to receive images and return recognized birthdates.
- Simplified interface without a graphical user interface.
- Scalability and accessibility for integration into various applications.

The deployment marks the transformation of the model into a practical tool with real-world applications.

Results

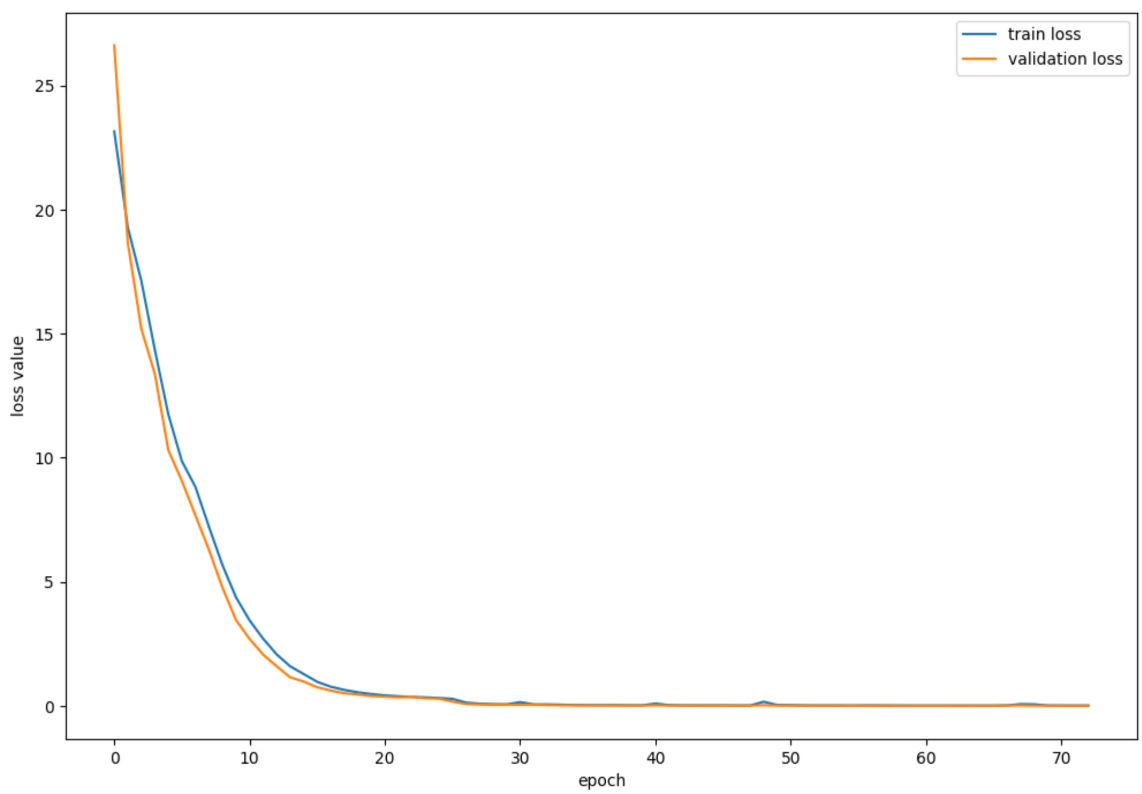
In this section, we will present the outcomes and achievements of the "OCR Arabic Birthdates" project, which culminated in a remarkable 100% accuracy score.

1. Model Training and Validation

   The model was meticulously trained on the prepared dataset, which encompassed a wide array of Arabic birthdates.During the training process, the model demonstrated exceptional learning capabilities, leading to a 100% accuracy score on the validation dataset. This result underscored the model's capacity to recognize Arabic birthdates accurately.

2. Training Loss Graph

   Below is a training loss graph that visually represents the model's learning progress over time:



3. Model Evaluation Metrics

Beyond accuracy, the model's performance was rigorously assessed using additional metrics, including:

- Character-Level Accuracy: OCR models can make errors in individual characters. Calculate the percentage of correctly predicted characters.
- Precision: Measuring the model's ability to correctly identify positive cases (Arabic birthdates) without false positives.
- Recall: Evaluating the model's capability to capture all positive cases in the dataset.
- F1-Score: Providing a balance between precision and recall, offering a comprehensive view of the model's performance.

```
Accuracy: 1.00

Precision: 1.00

Recall: 1.00

F1-Score: 1.00
```

```
correct char: 19980

total char: 19980

accuracy char: 100.00 %

______________________________

correct date: 1998

total date: 1998

accuracy date: 100.00 %
```