Document Classification and Data Extraction

ABSTRACT

We proposed a model that takes a pdf or image consisting of multiple documents and identifies the set of documents present in the pdf. This is done by splitting the input PDF into single pages. Each individual page is classified into its respective document type using the CNN model. Then we leverage OCR (optical character recognition) to extract data from each document. This is proposed for five documents (Aadhar, PAN, driver's license, passport, and voter ID). The input pdf must have a single document on one page, except for the front and back of the same document.
With respect to gains of `0.6923` and losses of `0.8340`, our data classification model achieved `0.7342` accuracy on the training set and `0.7736` accuracy on the validation set.

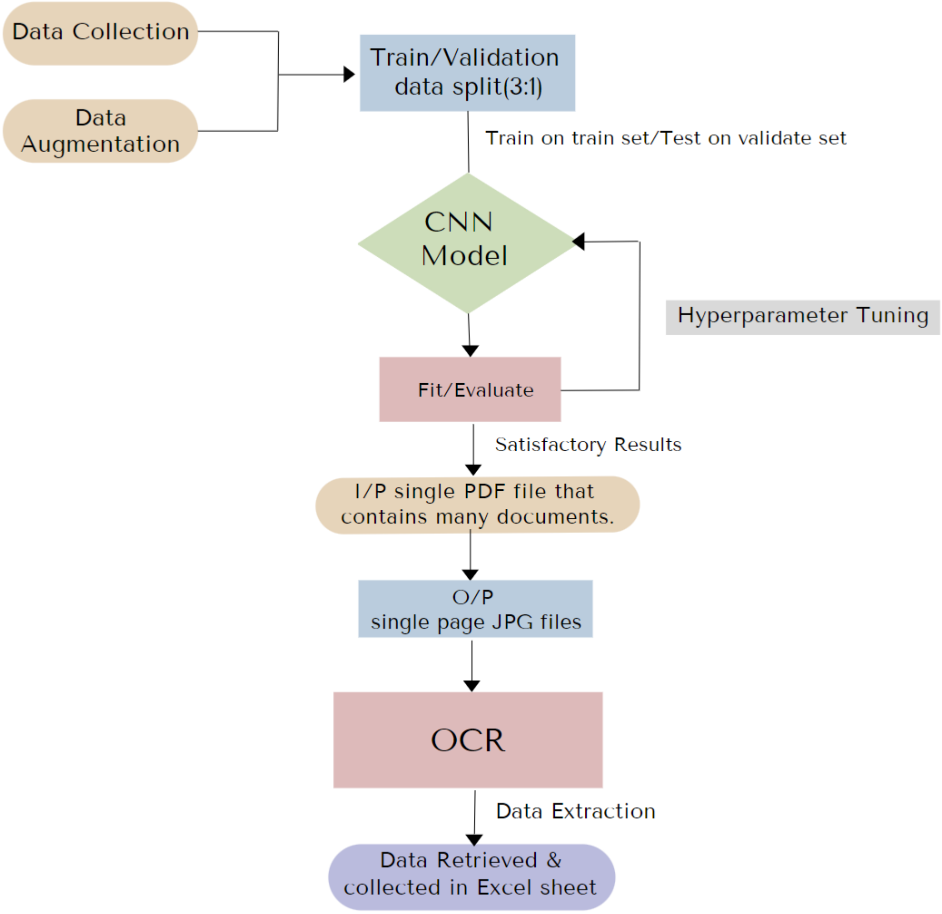
Data Collection
Data Augmentation
Train/Validation data split(3:1)
Train on train set/Test on validate set
CNN Model
Hyperparameter Tuning
Fit/Evaluate
Satisfactory Results
I/P single PDF file that contains many documents.
O/P single page JPG files
OCR
Data Extraction
Data Retrieved & collected in Excel sheet

Fig 1- Methodology

1. Data Collection

When we began searching for an appropriate dataset, we observed that there is no publicly available dataset of identity documents as they hold sensitive and personal information. But we came across a dataset on Kaggle that consisted of six folders, i.e., Aadhar Card, PAN Card, Voter ID, single-page Gas Bill, Passport, and Driver's License. We added a few more images to each folder. These were our own documents that we manually scanned, with the rest coming from Google Images. Later, we also added images from Google Drive provided by ecell-IITB. RGB and black-and-white images are present in all document classes.

Our final dataset looked as follows:

- Aadhar card(both front and back)-straight,rotated and having hands in background
- PAN card
- Voter ID (front and back)-with and without background
- Passport
- Driver's License-both rotated and straight

Thus, these are the **five documents** we are classifying and extracting information from.

Document Type	Adhaar card	PAN card	Driving license	Voter ID	Passport
No of images	48	41	64	76	19

Table 1 - Dataset Description

2. Data preprocessing-

Before model training, we applied horizontal and vertical data augmentation using random flips. This further increased the size and diversity of the dataset. The categorical values of the labels column were converted to numerical values using one-hot encoding.

3. Document Classification Model-

3.1 Training model

Before training, the model was built. The loss function was taken as categorical cross-entropy, and Adam was chosen as the optimizer. The data was split into 3:4 for training and 1:4 for validating. Early stopping (a form of regularization to avoid overfitting) was also implemented.

3.2 Hyperparameter Tuning

We discovered that our model is overfitting after evaluating it on the validation set, as the validation accuracy was significantly lower than the train accuracy. Our model demanded that we tune parameters to increase train accuracy. Various hyperparameters like the number of layers, neurons in each layer, number of filters, kernel size, value of p in dropout layers, number of epochs, batch size, etc. were changed until satisfactory training and validation accuracy was achieved.

Hyperparameters Tuned	Train Loss	Train Accuracy	Validation Loss	Validation Accuracy
Kernel size (3,3)	0.6801	0.6944	0.6723	0.8333
Activation function (Sigmoid)	0.5914	0.6944	0.5338	0.8333
Dropout (0.05)	1.0923	0.5190	0.9605	0.6415
Dropout (0.07)	0.5731	0.7342	0.7499	0.7736
Batch size (64)	1.3401	0.4000	1.4268	0.2500
Batch size (128)	1.4360	0.3385	1.4426	0.3409

Table 2: Hyperparameter Tuning

The following two graphs show the change in model accuracy (Fig. 2) and model loss (Fig. 3) on a set of hyperparameters against each epoch.

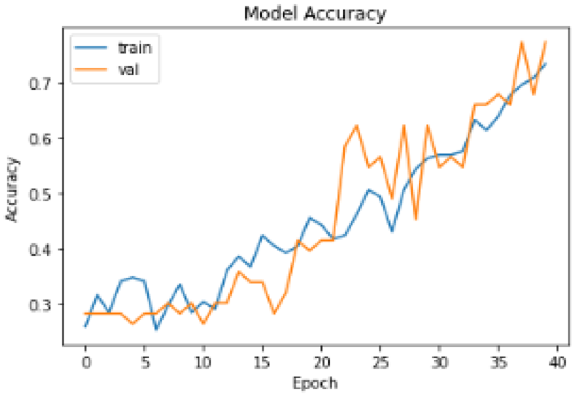

Fig. 2 : Train and Validate Accuracy at each Epoch

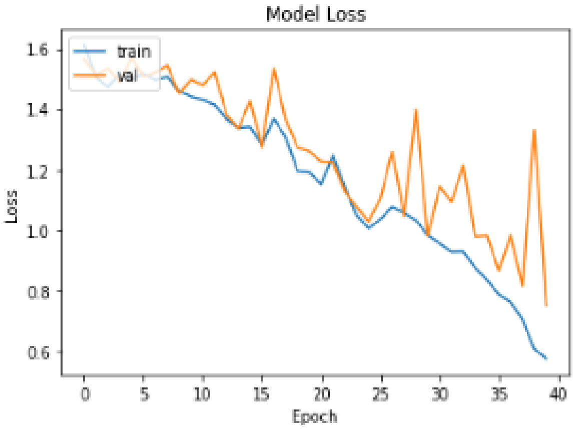

Fig. 3: Train and validation loss at each epoch

3.3 Choosing the final model

The model that was finally saved has the following evaluation metrics and hyperparameters.

No. of layers	Value of p in dropout	No. of epochs	Batch size	Kernel size	Activation function in first layer
9	0.07	40	32	3	ReLu

Table 3-Model's hyperparameters

Train Accuracy	0.7342
Validation Accuracy	0.7736

Table 4-Accuracy

Train Loss	0.6923
Validation Loss	0.8340

Table 5- Loss

4. Information extraction model (OCR)

OCR(Optical character recognition) enables text conversion of data from scanned documents. Following are the steps of OCR done on images:

1. Image Processing- First, we resize an image, and then we convert it to grayscale, which lowers an image to its most basic pixel. The image is then sharpened with a Laplacian filter. Then filter2D is applied on Image.
2. Text extraction- Using Pytesseract ,text was extracted from the images.

3. The acquired text and the class label are delivered to the appropriate classes. The routines are adhaar_read_data, pan_read_data, voterid_read_data, and driving_license_read, respectively, depending on whether the class label is aadhaar card, pan card, voter id card, or driving license.
4. Cleaning the extracted text -
    - Convert factor column to character columns
    - Remove white spaces in these columns
    - Removing special/unidentified characters
    - Convert the data back to factors
    - Leaving blank where the text is missing or has unusual values
5. Extraction using keywords- After the data cleaning process is completed, information was extracted. To discover attributes, specific keywords from the text were searched, for example, when finding voter numbers, keywords such as “ELECTION COMMISSION OF INDIA/CARD/IDENTITY CARD” are used.
6. Saving into excel- All the extracted information will be saved in a dictionary. This data will then be attached to a dataframe, which is then turned into a downloadable excel file.

References

1. Blogs:
    a. Document Classification With Machine Learning: Computer Vision, OCR, NLP, and Other Techniques: https://www.altexsoft.com/blog/document-classification/
    b. Deep Learning Based OCR for Text: https://nanonets.com/blog/deep-learning-ocr/
2. Leaderboards that are used to track progress in Document Classification: https://paperswithcode.com/task/document-classification
3. Dataset of identity documents: https://www.kaggle.com/datasets/omrastogi/identity-card-dataset
4. Research papers referred:
    a. https://arxiv.org/pdf/2106.04345.pdf
    b. .https://pub.inf-cv.uni-jena.de/pdf/Simon15:FCI.pdf
    c. https://link.springer.com/chapter/10.1007/978-3-319-68548-9_55#Bib1