Playing Cards image classification

1. Introduction

This problem is an image classification problem where we need to classify images of playing cards into 53 different labels (52 standard playing cards + 1 random). There was 7643 train images, 265 validation images and 265 test cards. Each image is 224x224 RGB image. I have used custom convolutional network as well as pretrained networks of ResNet18 and ResNet34.

Exploratory Data Analysis
All images in the dataset are high quality images with size 224x224x3. All images are cropped so that the card occupies more than 50% of the pixels in overall image. There are 7624 training images, 265 test images and 265 validation images. Each image of size 224x224x3.

2. Related Work

53 classes 7624 train, 265 test, 265 validation images 224 X 224 X 3 jpg format

About Dataset

This is a very high quality dataset of playing card images. All images are 224 X 224 X 3 in jpg format. All images in the dataset have been cropped so that only the image of a single card is present and the card occupies well over 50% of the pixels in the image. There are 7624 training images, 265 test images and 265 validation images. The train, test and validation directories are partitioned into 53 sub directories , one for each of the 53 types of cards. The dataset also includes a csv file which can be used to load the datasets.

14card types-14-(200 X 200)-94.61.h5(135.43 MB)

About this file

this is an EfficientNet B3 trained model. It has trained on 14 classes, one class for each rank of card (Ace,King,Quen etc). The model was trained on 200 X 200 X 3 image size. It achieved an F1 score of 94,61%. Note EfficientNet models expect pixels in the range of 0 to 255 so do NOT scale your images if you use this model.

Data Explorer

431.79 MB

folder

test

folder

train

folder

valid

- insert_drive_file

  14card types-14-(200 X 200)-94.61.h5
- insert_drive_file

  53cards-53-(200 X 200)-100.00.h5
- calendar_view_week

  cards.cs

What are Dataset Cards?

Each dataset may be documented by the `README.md` file in the repository. This file is called a **dataset card**, and the Hugging Face Hub will render its contents on the dataset's main page. To inform users about how to responsibly use the data, it's a good idea to include information about any potential biases within the dataset. Generally, dataset cards help users understand the contents of the dataset and give context for how the dataset should be used.

You can also add dataset metadata to your card. The metadata describes important information about a dataset such as its license, language, and size. It also contains tags to help users discover a dataset on the Hub, and data files configuration options. Tags are defined in a YAML metadata section at the top of the `README.md` file.

Dataset card metadata

A dataset repo will render its README.md as a dataset card. To control how the Hub displays the card, you should create a YAML section in the README file to define some metadata. Start by adding three --- at the top, then include all of the relevant metadata, and close the section with another group of --- like the example below:

The metadata that you add to the dataset card enables certain interactions on the Hub. For example:

- Allow users to filter and discover datasets at https://huggingface.co/datasets.
- If you choose a license using the keywords listed in the right column of this table, the license will be displayed on the dataset page.

When creating a README.md file in a dataset repository on the Hub, use Metadata UI to fill the main metadata:

Uploading datasets

The Hub is home to an extensive collection of community-curated and research datasets. We encourage you to share your dataset to the Hub to help grow the ML community and accelerate progress for everyone. All contributions are welcome; adding a dataset is just a drag and drop away!

Start by creating a Hugging Face Hub account if you don't have one yet.

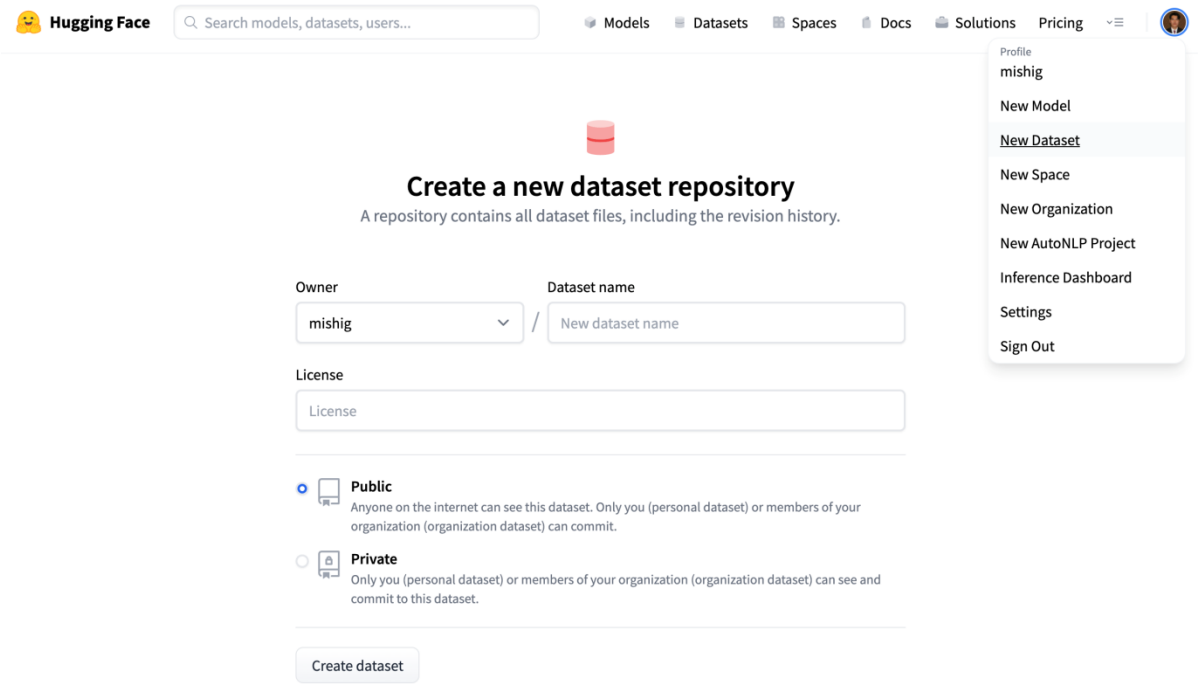
Upload using the Hub UI

The Hub's web-based interface allows users without any developer experience to upload a dataset.

Create a repository

A repository hosts all your dataset files, including the revision history, making storing more than one dataset version possible.

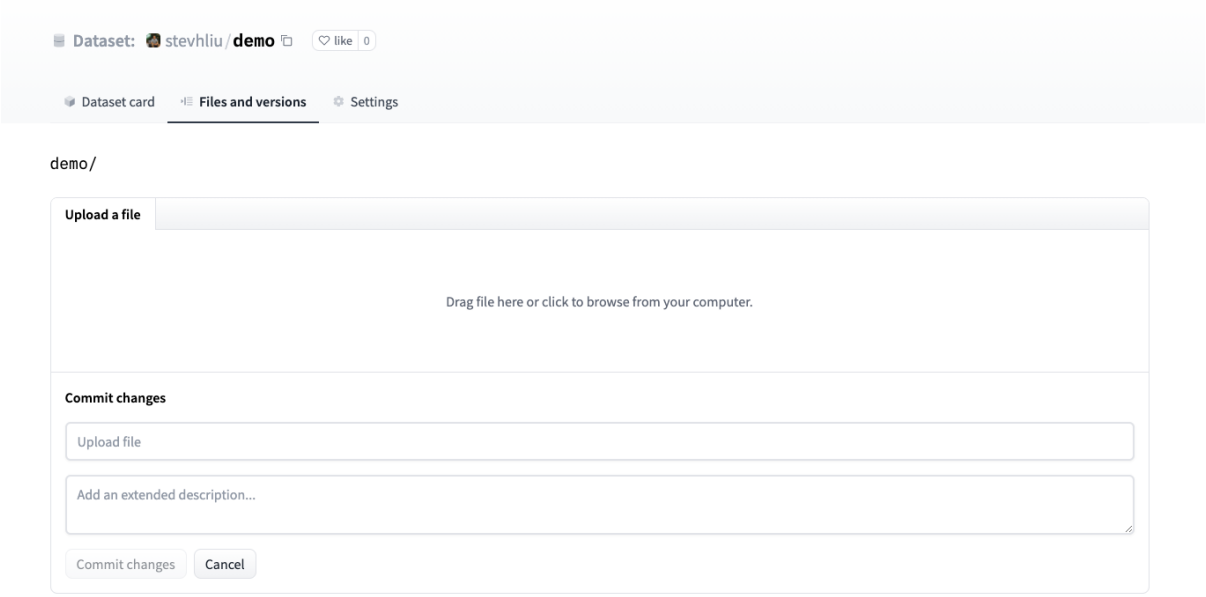
1. Click on your profile and select **New Dataset** to create a new dataset repository.
2. Pick a name for your dataset, and choose whether it is a public or private dataset. A public dataset is visible to anyone, whereas a private dataset can only be viewed by you or members of your organization.



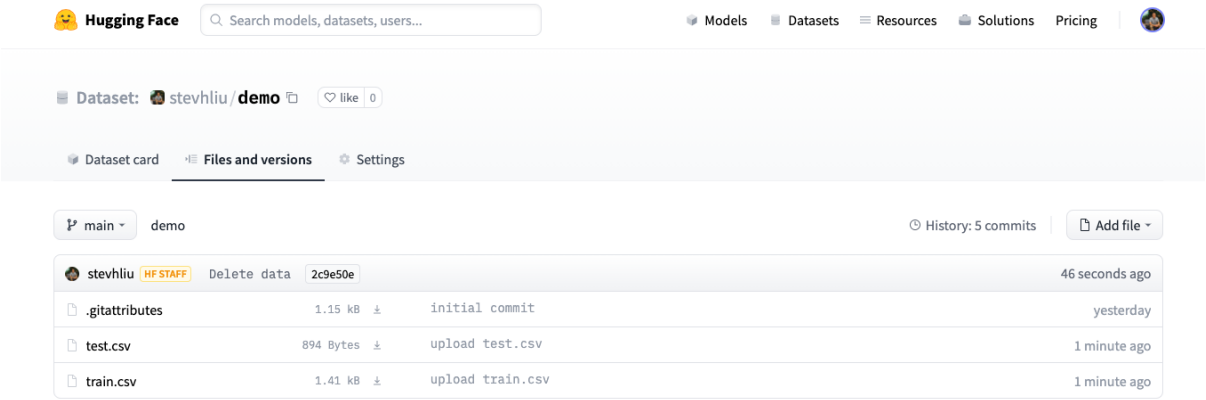
Upload dataset

1. Once you've created a repository, navigate to the **Files and versions** tab to add a file. Select **Add file** to upload your dataset files. We support many text, audio, image and other data extensions such as .csv, .mp3, and .jpg (see the full list of File formats).

2. Drag and drop your dataset files.



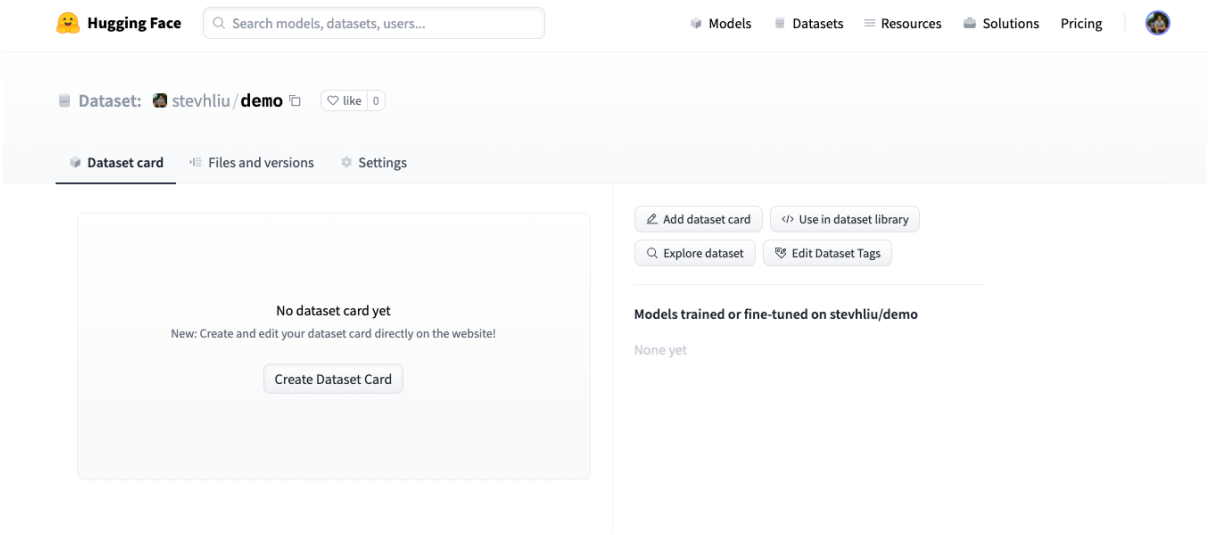
3. After uploading your dataset files, they are stored in your dataset repository.



Create a Dataset card

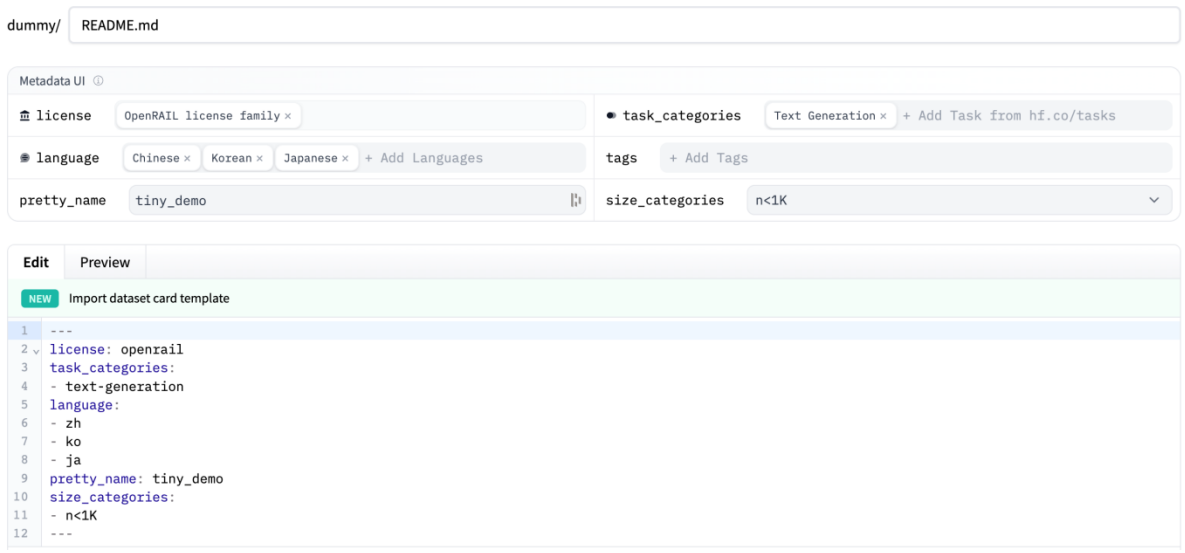
Adding a Dataset card is super valuable for helping users find your dataset and understand how to use it responsibly.

1. Click on **Create Dataset Card** to create a Dataset card. This button creates a README.md file in your repository.



2. At the top, you'll see the **Metadata UI** with several fields to select from such as license, language, and task categories. These are the most important tags for helping users discover your dataset on the Hub (when applicable). When you select an option for a field, it will be automatically added to the top of the dataset card.

   You can also look at the Dataset Card specifications, which has a complete set of allowed tags, including optional like `annotations_creators`, to help you choose the ones that are useful for your dataset.



3. Write your dataset documentation in the Dataset Card to introduce your dataset to the community and help users understand what is inside: what are the use cases and limitations, where the data comes from, what are important ethical considerations, and any other relevant details.

You can click on the **Import dataset card template** link at the top of the editor to automatically create a dataset card template. For a detailed example of what a good Dataset card should look like, take a look at the CNN DailyMail Dataset card.

Using the huggingface_hub client library

The rich features set in the huggingface_hub library allows you to manage repositories, including creating repos and uploading datasets to the Hub. Visit the client library's documentation to learn more.

Using other libraries

Some libraries like 🤗 Datasets, Pandas, Polars, Dask or DuckDB can upload files to the Hub. See the list of Libraries supported by the Datasets Hub for more information.

Using Git

Since dataset repos are Git repositories, you can use Git to push your data files to the Hub. Follow the guide on Getting Started with Repositories to learn about using the git CLI to commit and push your datasets.

File formats

The Hub natively supports multiple file formats:

- CSV (.csv, .tsv)
- JSON Lines, JSON (.jsonl, .json)
- Parquet (.parquet)
- Arrow streaming format (.arrow)
- Text (.txt)
- Images (.png, .jpg, etc.)
- Audio (.wav, .mp3, etc.)
- WebDataset (.tar)

It supports files compressed using ZIP (.zip), GZIP (.gz), ZSTD (.zst), BZ2 (.bz2), LZ4 (.lz4) and LZMA (.xz).

Image and audio files can also have additional metadata files. See the Data files Configuration on image and audio datasets, as well as the collections of example datasets for CSV, TSV and images.

You may want to convert your files to these formats to benefit from all the Hub features. Other formats and structures may not be recognized by the Hub.

Which file format should I use?

For most types of datasets, Parquet is the recommended format due to its efficient compression, rich typing, and since a variety of tools supports this format with optimized read and batched operations. Alternatively, CSV or JSON Lines/JSON can be used for tabular data (prefer JSON Lines for nested data). Although easy to parse compared to Parquet, these formats are not recommended for data larger than several GBs. For image and audio datasets, uploading raw files is the most practical for most use cases since it's easy to access individual files. For large scale image and audio datasets streaming, WebDataset should be preferred over raw image and audio files to avoid the overhead of accessing individual files. Though for more general use cases involving analytics, data filtering or metadata parsing, Parquet is the recommended option for large scale image and audio datasets.

Dataset Viewer

The Dataset Viewer is useful to know how the data actually looks like before you download it. It is enabled by default for all public datasets. It is also available for private datasets owned by a PRO user or an Enterprise Hub organization.

After uploading your dataset, make sure the Dataset Viewer correctly shows your data, or Configure the Dataset Viewer.

3. Materials and Experimental Evaluation

3.1 Dataset

The dataset used for this project is the Cards Image Dataset-Classification, which contains over 8000 images of poker cards in JPG format. The dataset is divided into three sets: training (7624 images), validation (265 images), and testing (265 images). The images are of size 224 x 224 pixels with three color channels.

Methodology

The project follows a standard machine learning workflow, including the following steps:

1. **Data Collection:** The dataset was downloaded from Kaggle and examined to gain insights into the data.
2. **Data Preprocessing:** The images were preprocessed by resizing them to a standard size and converting them into an array format suitable for feeding into our CNN model.
3. **Data Augmentation:** Data augmentation techniques were used to increase the size of the dataset and improve the robustness of the model. Techniques used include image rotation, flipping, and zooming.
4. **Model Building:** A CNN model was built using Keras with TensorFlow backend, and trained using the preprocessed dataset. Different architectures, hyperparameters, and optimization algorithms were experimented with to achieve optimal performance.
5. **Model Evaluation:** The performance of the model was evaluated using various metrics such as accuracy, precision, recall, and F1 score. The results were visualized using a confusion matrix and classification report.
6. **Model Deployment:** The trained model was deployed to make predictions on new, unseen poker card images. The predictions were visualized.

Getting Started

To run this project, you will need to have the following tools and libraries installed:

- Python 3.8+
- Keras
- TensorFlow
- NumPy
- Matplotlib
- Scikit-learn

You can run the project on your local machine or in a Jupyter notebook environment such as Google Colab.

1. Clone this repository to your local machine:

git clone https://github.com/yourusername/poker-card-image-recognition.git

2. Download the dataset from Kaggle and extract it to the data folder in the project directory.
3. Open the Jupyter notebook poker-card-image-recognition.ipynb and run the cells in order.
4. Once the model is trained, you can use it to make predictions on new poker card images.

3.2 Methodology

What specific hypotheses does your experiment test? What are the criteria you are using to evaluate your method? Describe the experimental methodology that you used. What is the training/test data that was used? Exactly what performance data did you collect and how are you presenting and analysing it? (please link your Multi Line plot)

3.3 Results

Present the quantitative results of your experiments. What are the basic differences revealed in the data.Comparisons to competing methods that address the same problem are particularly useful.(please link your confusion matrix and classification report)

Example :

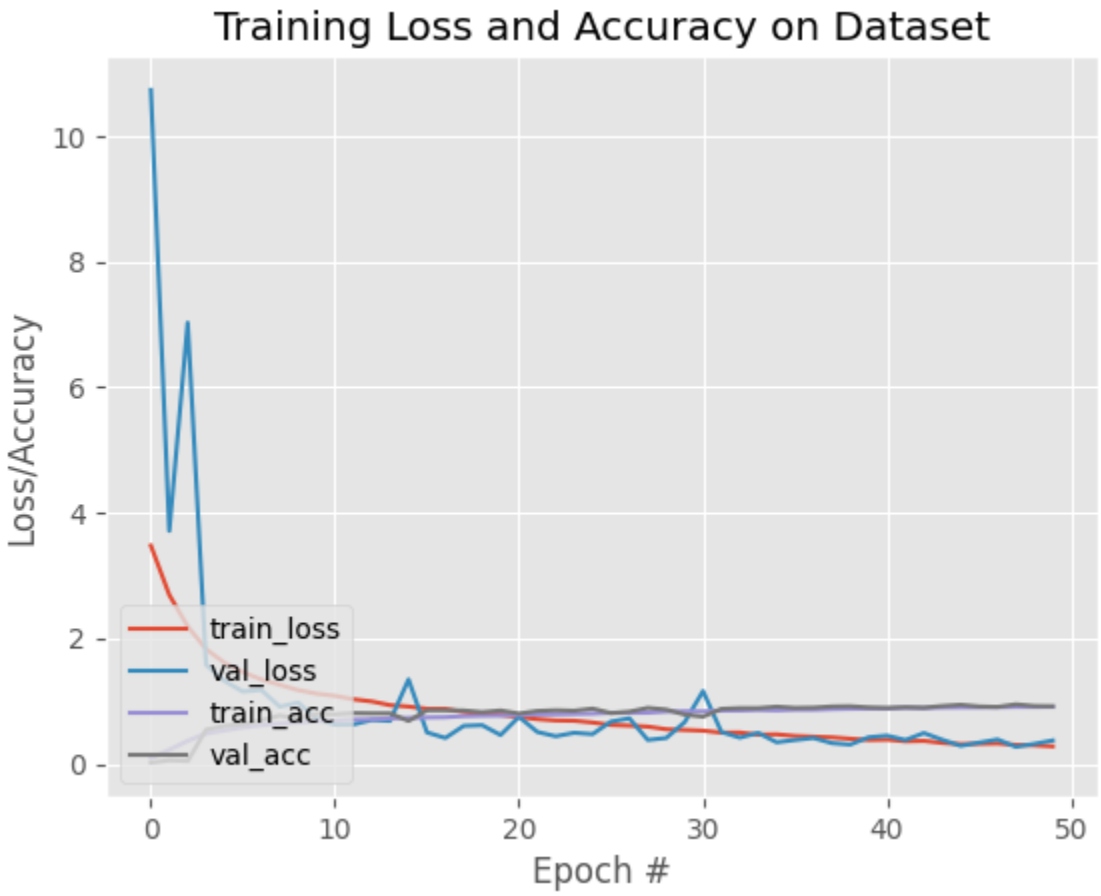


Figure 1: Confusion Matrix

3.4 Discussion

What conclusions do the results support about the strengths and weaknesses of your method compared to other methods? How can the results be explained in terms of the underlying properties of the algorithm and/or the data.

4. Future Work

What are the major shortcomings of your current method? For each shortcoming, propose additions or enhancements that would help overcome it.

5. Conclusion

Briefly summarize the important results and conclusions presented in the paper. What are the most important points illustrated by your work? How will your results improve future research and applications in the area?

6.Reference

Be sure to include a standard, well-formatted, comprehensive bibliography with citations from the text referring to previously published papers in the scientific literature that you utilised or are related to your work.(any format)

Reference Example :

1. Kwak G-H, Park N-W. Impact of Texture Information on Crop Classification with Machine Learning and UAV Images. *Applied Sciences*. 2019; 9(4):643. https://doi.org/10.3390/app9040643
2. Dataset Source :