Japanese Handwriting Image Recognition

Author: Chaz Frazer

Table of Contents

Questions

- How do we use this data to improve image recognition, especially with archival handwritten data that needs to be transcribed?
- Can this model be used to create enhanced user interfaces and APIs with touchscreen devices?


少なくとも二つの言語を理解するまで、決して一つの言語も理解することはない。
— You can never understand one language, until you understand at least two.

二

桜

Data & Info

- Data from the National Institute of Advanced Industrial Science and Technology (AIST)
- Reorganized by the Japan Electronics and Information Technology Industries Association
- About 1.2 million handwritten Japanese records including numerals, hiragana, katakana, and kanji
- Data collected from 1973 to 1984
- Data was collected by submission of magnetic tapes and CD-R delivered by post

What are Japanese Characters?

あ

Hiragana

Main phonetic alphabet. Used for mainly for context and connecting sentence syntax

漢字

Kanji

Borrowed Chinese characters. Used for major parts of speech

ア

Katakana

Phonetic alphabet used for borrowed foreign words, onomatopoeia, and sounds

The Data

Hiragana:

- 71 hiragana characters (46 unique + 29 diphthongs
- 160 writers
- 8199 records (each record has 10 sheets, each writer wrote 956 characters
- 1,254,120,000 unique handwritten characters in dataset

Kanji:

- 883 daily use kanji (3000 for publications, 35,000+ exist)
- 160 writers
- 8199 records
- 152,878,411 unique handwritten kanji in dataset

Katakana:

- 46 katakana characters (+ same amount of diphthongs as hiragana)
- 1411 writers
- 2052 records
- 2,436,366 unique handwritten katakana characters in the dataset

Data Cleaning & Import

- Data was read in from binary, sorted, and saved to an npz file to be re-read.
- Images reshaped to 48*48 pixels
- Hiragana & Kanji shared a dataset, so they needed to be differentiated
- Data labels created and images visualized from binary for accurate import

Handwriting Images

Modeling & Prep

- Variation in dataset created by using ImageDataGenerator (rotated, zoomed, & shifted). This created enough variation to avoid overfitting. This is a submodule of tensorflow.keras
- CNN model used to fit the data (I used my PC's GPU but still had to run my kanji model in particular overnight and pray!)
- Tensorboard used to capture logs of the kanji modeling process for later visualization
- Activations used: ReLu, Softmax
- Optimizers: Adam
- Loss Function: Sparse categorical cross entropy

ImageDataGenerator

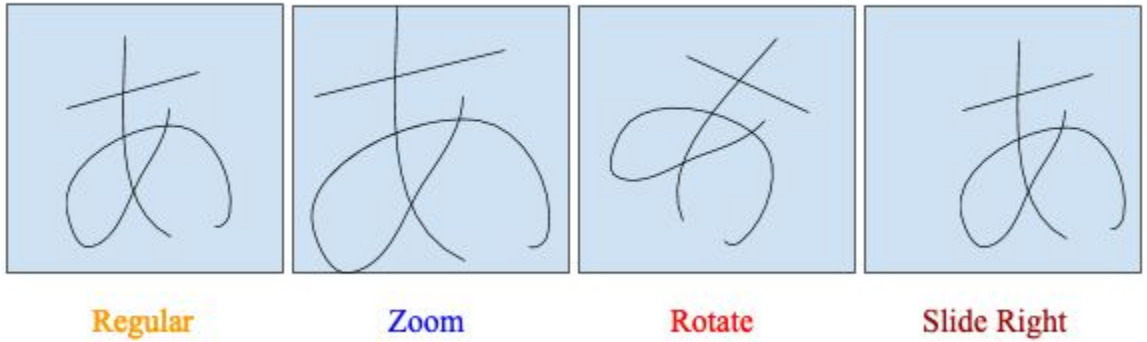


Image % Aiyu Kamate

Model Parameters

Early Stopping

Stop training if accuracy doesn't improve after specified epochs

ReduceLRonPlateau

Reduce learning rate if accuracy doesn't improve (specifiable)

Image Manipulation

ImageDataGenerator, Datagen.flow (to fit the model)

Layers

Max Pooling, Dropout, Dense

Tensorboard

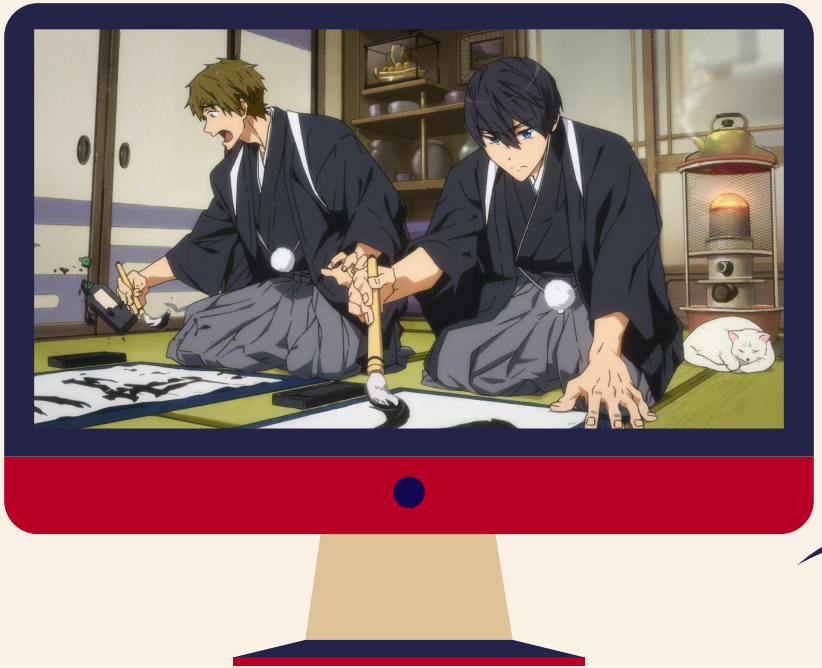
Callback to monitor model performance in real-time

Save Model

Model saved using .h5 format

Conclusions & Results

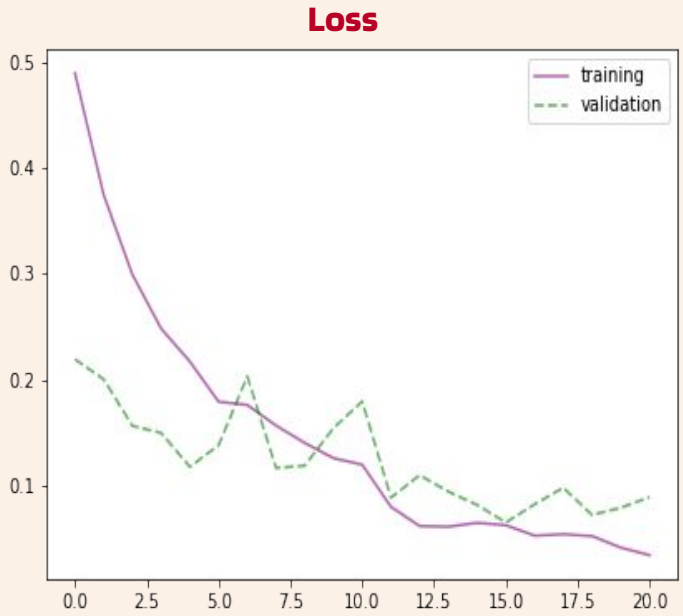
四

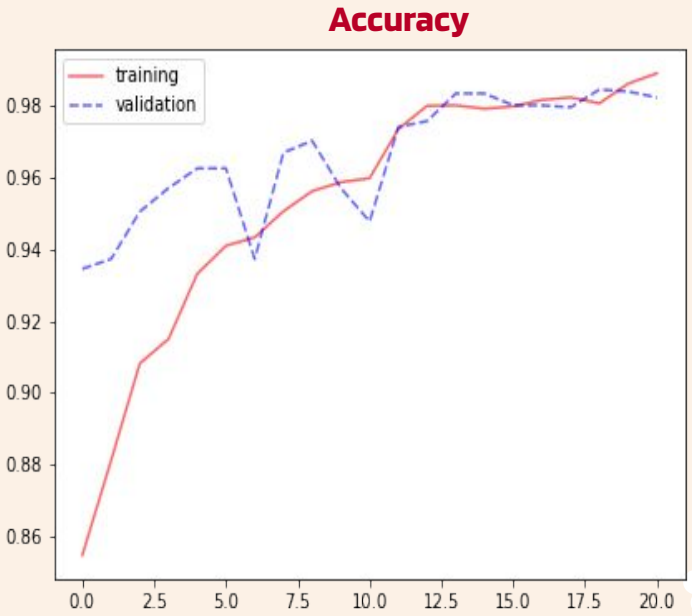


Model Results

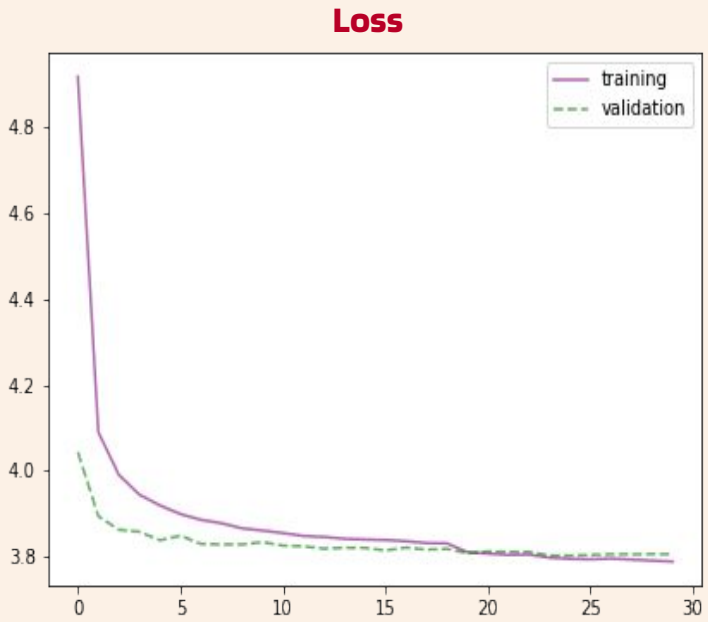
	Hiragana	Kanji	Katakana
Training	98.91%	39.49%	98.90%
Cross-val	98.24%	39.28%	98.93%
Test Data	98.33%	39.44%	98.78%

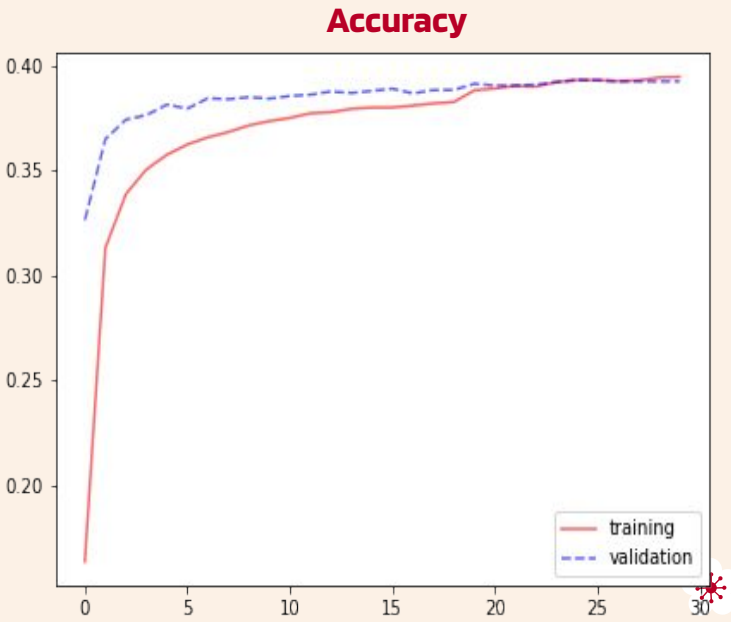
Hiragana Loss vs Accuracy



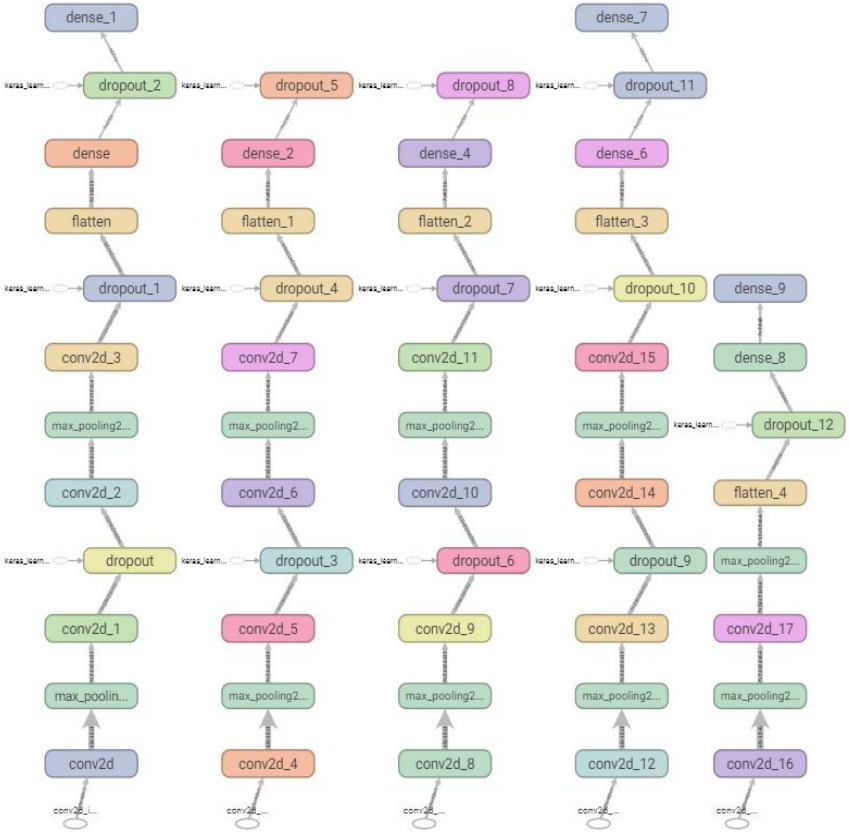


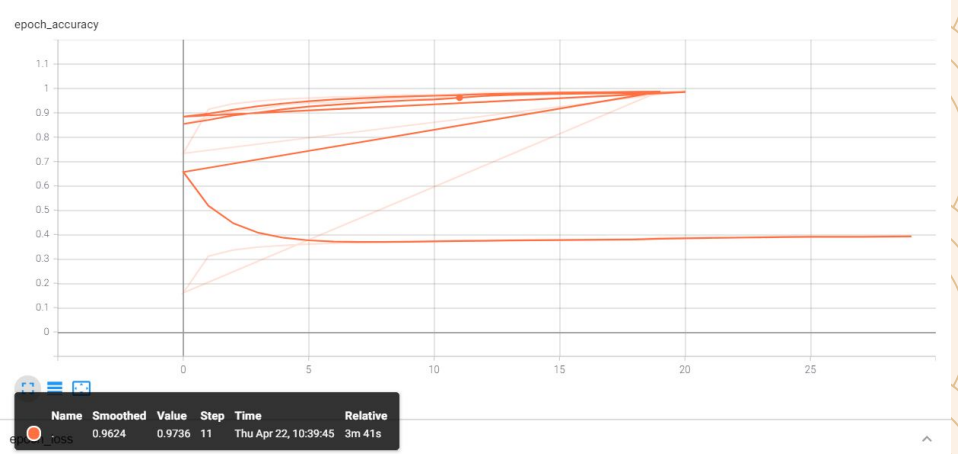
Kanji Loss vs Accuracy

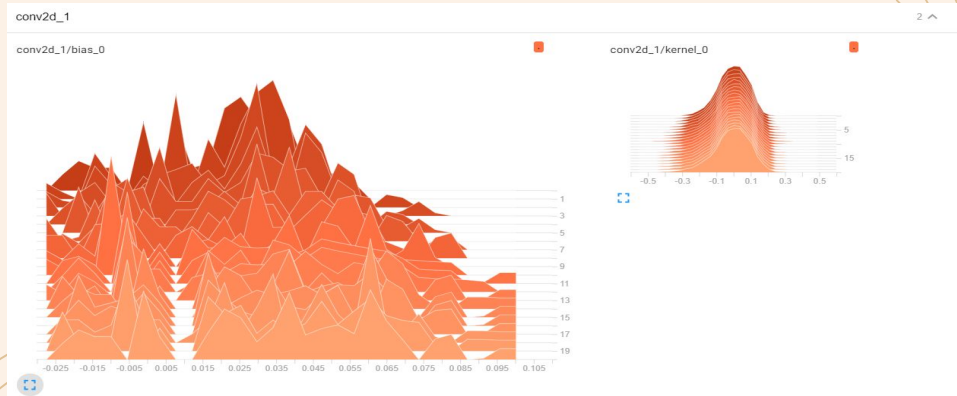




Tensorboard Visuals







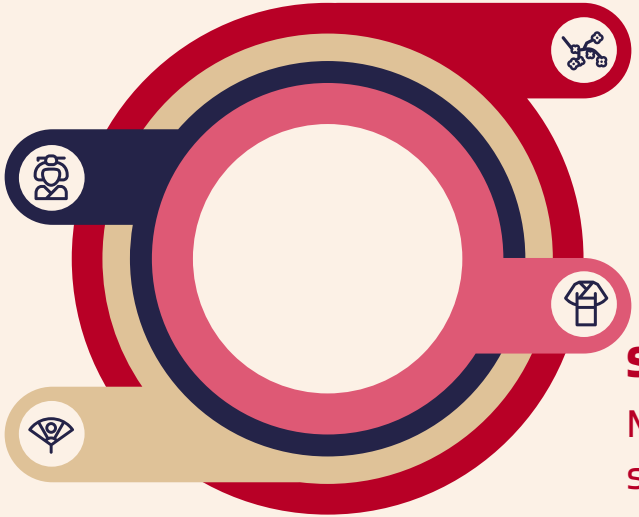
What is up with that Kanji Model?

Data Variation

883 separate classes of data for modeling

Rework Params

Need to rework parameters and research proven approaches. Will re-evaluate



Radicals

Kanji are composed in "parts" called radicals. These radicals are also the basis for many original hiragana & katakana characters

Similarity

Many kanji are quite similar to each other, especially with radicals combinations that express similar ideas

Next Steps

Expand model for touchscreen API integration for language education

Fine tune model params for kanji model to achieve higher accuracy

Implement cloud-based modeling using other model techniques

Thanks!

Github.com/Mynusjanai
Mynus.gg

Dataset: http://etlcdb.db.aist.go.jp/download-request