Practical machine learning and deep learning course project deliverable 3

Automated manga cleaning using CNNs

Team:
Vyacheslav Sergeev
vy.sergeev@innopolis.university
Nikita Strygin
n.strygin@innopolis.university
Dinislam Gabitov
d.gabitov@innopolis.university

Summary of project focused efforts

During last two weeks we have managed to:
1. Investigate photoshop plugin construction
2. Complete clean up the data that we had
3. Augment the dataset

The following passages describe our efforts in more detail.

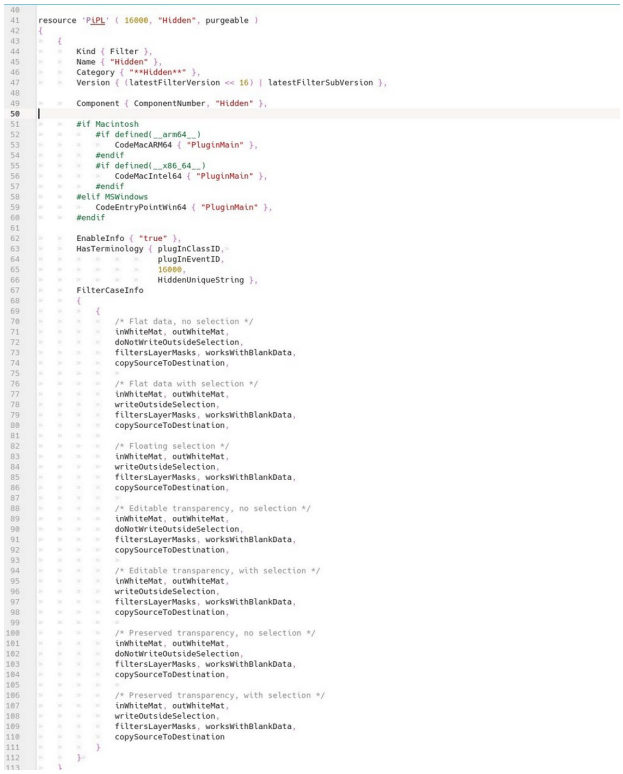
The photoshop plugin

Our main effort was spent investigating how one could construct a plugin for the photoshop. There are 3 APIs that are avalable for usage. However one of them is line for deprecation, so we researched mainly 2 of them.

First one is UXP API. It is in javascript, has a modern feel to it and utilizes rather common toolchain. UXP allows for fancy interface manipulation and is easy to use. However it does not allow to modify content e.g. pictures, layers etc. in any way – there is just no API to access that. Our speculation is that it is slow to bring that content to the javascript.

The second one is C API. The C API for the Photoshop is the complete opposite of the the UXP API. The API was designed in the 90s and it feels exactly its age. It is complicated, it has a lot of its own unique definitions and data/resource formats. Its documentation is quite convoluted as well. Now we know the reason why there are not tons of hugely popular photoshop plugins available for usage. However it has the main part that we need – access to page content, so we have no choice and will have to get acquainted with this API sooner or later.

In the end It is great that we decided to start on this part a bit early as finding out about the C API later could have killed the idea alltogether. However we will have to see how well the API works and whether we can scrap together enough to make plugin happen.



```
resource 'PiPL' ( 16000, "Hidden", purgeable )
{
    {
        Kind { Filter },
        Name { "Hidden" },
        Category { "**Hidden**" },
        Version { (latestFilterVersion << 16) | latestFilterSubVersion },

        Component { ComponentNumber, "Hidden" },

        #if Macintosh
            #if defined(__arm64__)
                CodeMacARM64 { "PluginMain" },
            #endif
            #if defined(__x86_64__)
                CodeMacIntel64 { "PluginMain" },
            #endif
        #elif MSWindows
            CodeEntryPointWin64 { "PluginMain" },
        #endif

        EnableInfo { "true" },
        HasTerminology { plugInClassID,
                         plugInEventID,
                         16000,
                         HiddenUniqueString },
        FilterCaseInfo
        {
            {
                /* Flat data, no selection */
                inWhiteMat, outWhiteMat,
                doNotWriteOutsideSelection,
                filtersLayerMasks, worksWithBlankData,
                copySourceToDestination,

                /* Flat data with selection */
                inWhiteMat, outWhiteMat,
                writeOutsideSelection,
                filtersLayerMasks, worksWithBlankData,
                copySourceToDestination,

                /* Floating selection */
                inWhiteMat, outWhiteMat,
                writeOutsideSelection,
                filtersLayerMasks, worksWithBlankData,
                copySourceToDestination,

                /* Editable transparency, no selection */
                inWhiteMat, outWhiteMat,
                doNotWriteOutsideSelection,
                filtersLayerMasks, worksWithBlankData,
                copySourceToDestination,

                /* Editable transparency, with selection */
                inWhiteMat, outWhiteMat,
                writeOutsideSelection,
                filtersLayerMasks, worksWithBlankData,
                copySourceToDestination,

                /* Preserved transparency, no selection */
                inWhiteMat, outWhiteMat,
                doNotWriteOutsideSelection,
                filtersLayerMasks, worksWithBlankData,
                copySourceToDestination,

                /* Preserved transparency, with selection */
                inWhiteMat, outWhiteMat,
                writeOutsideSelection,
                filtersLayerMasks, worksWithBlankData,
                copySourceToDestination
            }
        };
    }
```

Example of resource format for C Photoshop API

Data cleanup

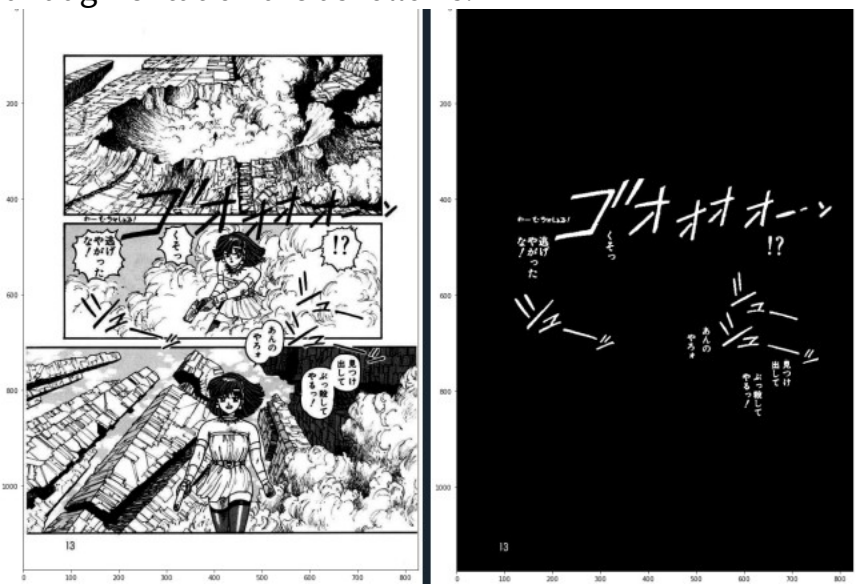
As we have found out before, the data had some inaccuracies in labeling. This time we have cleaned up the data for good removing hopefull all of bad samples. While our dataset is rather small it still took ~3 hours to comb through all of it comparing the labels with the data. However this means that we can rely on the data and augment it in any way we want.

Example of the badly labeled data
(missing a few SFX)

Data augmentation

In order to compensate for rather small dataset size we decided to do data augmentation. One of the first ideas came from the source articles – we decided to crop parts of images to augment the dataset. Another idea is rather original and involves more of a dataset construction. We take a manga image and generate some random japanese text in multitude of styles and fonts on top of the manga. If done correctly it will allow our model to generalise better. However the approach is not without its difficulties. First we had to figure out which text to put into images. Letters in random order can possibly distract the model from identifying text correctly as they would have different patterns, so we have opted to search for a Lorem Ipsum generator. Lorem Ipsum was originally a scrambled book in english that appeared in 1914. Luckily for the book it was perfect for testing text layouts as it was really similar to normal english in word patterns, but the reader would not be distracted by the meaning of the text. We have found the same thing but for japanese located at https://lipsum.sugutsukaeru.jp/index.cgi

The results for our augmentation are as follows:



Original image example

Random crop augmentation example

Random text augmentation example