Comparison of OCR and CNN Classification Results

Ryan Heslin

April 20, 2022

This document compares the results of classifying all images, using both a trained neural network and optical character recognition with `tesseract`.

```
source(here::here("R", "classifier_utilities.R"))
ocr_results <- readRDS(here::here("data", "classifier", "outputs", "ocr_results.Rds"))
cnn_results <- read.csv(here::here("data", "classifier", "outputs", "cnn_results.csv"))
latest_date <- max(cnn_results[["date"]])
cnn_results <- read.csv(here::here("data", "classifier", "outputs", "cnn_all_classifications.csv"))
```

I opted to record results separately, since I planned to record CNN results many times. The datasets can be joined using `image` name as a key. The columns of `cnn_results` are:

- `actual_class`, `pred_class`: True and predicted classes, with 1 corresponding to non-Trump and 2 to Trump
- `image`: Path to the image corresponding to the row
- `p_trump`: Predicted Trump probability
- `log_p_trump`: Log of predicted Trump probability
- `batch_loss`: Value of loss function for batch the observation belonged to

The columns of `ocr_results` are: * `image` and `ocr_pred_class`: The same meanings as in `cnn_results` * `ocr_text`: The raw text recognized by `tesseract`. Prediction was done by case-insensitive matching for the string "trump" in this text. * `ocr_output`: List column of detailed `tesseract` results. Each entry contains the columns `word` (each distinct recognized word, concatenated to form `ocr_text`), `confidence` (percentage confident in word identification), and `bbox` (bounding box of image coordinates corresponding to identified word). Because this is a list column, I chose to store the data as an RDS.

```
combined <- merge(ocr_results, cnn_results, by = "image")
```

All images are in both result sets.

```
setequal(ocr_results$image, cnn_results$image)
```

```
[1] TRUE
```

Comparing Misclassifications

A contingency table shows the different classifications. Most agree.

```
with(combined, table(cnn = CLASS_NAMES[pred_class], ocr = CLASS_NAMES[ocr_pred_class])) %>%
  data.frame()
```

cnn	ocr	Freq
no_trump	no_trump	632
trump	no_trump	72
no_trump	trump	1
trump	trump	2

Here is a more detailed breakdown of predicted and actual labels. The network appeared to be more likely to predict an image contained Trump than OCR. I added training and loss weights to compensate for the shortage of Trump images, but I think they may need to be adjusted downward.

```
with(combined, table(cnn = CLASS_NAMES[pred_class], ocr = CLASS_NAMES[ocr_pred_class], true = CLASS_NAM
  knitr::kable()
```

cnn	ocr	true	Freq
no_trump	no_trump	no_trump	631
trump	no_trump	no_trump	36
no_trump	trump	no_trump	0
trump	trump	no_trump	0
no_trump	no_trump	trump	1
trump	no_trump	trump	36
no_trump	trump	trump	1
trump	trump	trump	2

In most cases, both classifiers were correct. Interestingly, the number of cases where only one was correct is about the same for both.

```
combined <- within(combined, prediction_category <- ifelse(actual_class == ocr_pred_class,
  ifelse(pred_class == actual_class, "both", "ocr"),
  ifelse(pred_class == actual_class, "cnn", "neither")
))
```

```
sort(table(combined[["prediction_category"]]), decreasing = TRUE) |>
  as.list() |>
  list2DF()
```

both	ocr	cnn	neither
633	37	36	1

One or both classifiers was wrong in 0.105 of cases. The overall error rate from naively predicting "not Trump" in every case would be 0.057

We can examine the images where at least one classification was incorrect. This plot shows the first four of those 74 images. I don't see any obvious pattern. `tesseract` seems to have correctly classified some images with no text, which I have to regard as spurious.

```
disagreements <- combined[["prediction_category"]] != "both"
to_plot <- seq_len(4)
inspect_images_from_paths(
  paths = combined[disagreements, "image"][to_plot],
  labels = paste("Correctly classified by:", combined[disagreements, "prediction_category"][to_plot]),
  plot_dims = c(2, 2)
)
```

area_25_republican_committee_bgimg_20220211.png

Correctly classified by: ocr

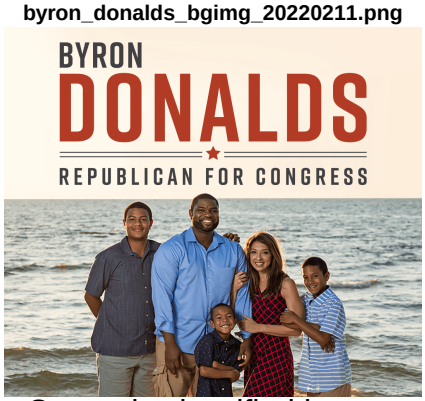
brian_for_better_nj_inc_bgimg_20220211.png

Correctly classified by: ocr

austin_chenge_for_governor_bgimg_20220211.jpg

Correctly classified by: ocr

byron_donalds_bgimg_20220211.png



Correctly classified by: ocr

Here are tables showing the true labels and predictions from both classifiers for the disagreements.

```
# Complicated code to wrangle a table
subset <- combined[disagreements, c(
  "image", "ocr_pred_class",
  "pred_class", "actual_class", "prediction_category"
)]
rownames(subset) <- NULL
subset[["image"]] <-
  tools::file_path_sans_ext(basename(subset[["image"]]))
subset[, endsWith(colnames(subset), "class")] <-
  lapply(
    subset[, endsWith(colnames(subset), "class")],
    function(x) CLASS_NAMES[x]
  )
subset[["image"]] <- gsub("_", "\n", subset$image) %>%
  lapply(strwrap, 20)
n_lines <- lengths(subset[["image"]])
subset[["image"]] <- lapply(
  subset[["image"]],
  function(x) {
    paste(
      "{\\shortstack[l]{",
      gsub("(?<!\\\\)\\s(?!\\\\)", "\\_",
        paste(x, collapse = " \\\\ "),
        perl = TRUE
      ),
      "}}"
    )
  }
)

col_spec <- paste0("|", paste(c("l", rep("c", ncol(subset) - 1)),
  collapse = "|"
), "|")
table_size <- 15
for (spl in split(subset, seq_len(nrow(subset)) %/% table_size)) {
  rows <- gsub("_", "\\\\_", paste(do.call(paste, append(spl, list(sep = " & "))),
    collapse = " \\\\\\n\\hline \n"
  ))


  display <- paste(
    paste0("\\noindent \\begin{tabular}{", col_spec, "}\n"),
    "\\toprule\n",
    "Image & OCR Prediction & CNN Prediction & True Class & Which Correct \\\\\\n",
    "\\hline\n",
    rows,
    "\\\\\\n",
    "\\bottomrule\n",
    "\\end{tabular}",
    collapse = "\n"
  )
  cat(display, "\n\r\n")
```

}

Image	OCR Prediction	CNN Prediction	True Class	Which Correct
area_25_republican committee_bgimg 20220211	no_trump	trump	no_trump	ocr
austin_chenge_for governor_bgimg 20220211	no_trump	trump	no_trump	ocr
brian_for_better_nj inc_bgimg_20220211	no_trump	trump	no_trump	ocr
byron_donalds_bgimg 20220211	no_trump	trump	no_trump	ocr
carlos_gimenez bgimg_20220211	no_trump	trump	no_trump	ocr
citizens_for_amy summers_bgimg 20220211	no_trump	trump	no_trump	ocr
commette_to_elect barbara_v_johnson for_new_mexico court_of_appeals bgimg_20220211	no_trump	trump	no_trump	ocr
committee_to_elect kerry_j_morris_to new_mexico_supreme court_bgimg 20220211	no_trump	trump	no_trump	ocr
committee_to_elect stephen_cicak_bgimg 20220211	no_trump	trump	no_trump	ocr
committee_to_elect thomas_ardinger bgimg_20220211	no_trump	trump	no_trump	ocr
douglas_county republican_party bgimg_20220211	no_trump	trump	no_trump	ocr
greg_lirette_bgimg 20220211	no_trump	trump	no_trump	ocr
james_risch_bgimg 20220211	no_trump	trump	no_trump	ocr
jasinski_for_senate bgimg_20220211	no_trump	trump	no_trump	ocr

Image	OCR Prediction	CNN Prediction	True Class	Which Correct
jerry_carl_bgimg 20220211	no_trump	trump	no_trump	ocr
john_barrasso_bgimg 20220211	no_trump	trump	no_trump	ocr
jordan_m_mackey_for ks_house_33rd_bgimg 20220211	no_trump	trump	no_trump	ocr
joshua_morris_for state representative bgimg_20220211	no_trump	trump	no_trump	ocr
julian_acciard bgimg_20220211	no_trump	trump	no_trump	ocr
karl_allred_for house_19_bgimg 20220211	no_trump	trump	no_trump	ocr
kevin_cramer_bgimg 20220211	no_trump	trump	no_trump	ocr
lindsey_graham bgimg_20220211	no_trump	trump	no_trump	ocr
luke_hellier_for city_council_bgimg 20220211	no_trump	trump	no_trump	ocr
matthew_burkhart bgimg_20220211	no_trump	trump	no_trump	ocr
mike_bost_bgimg 20220211	no_trump	trump	no_trump	ocr
milliman_for colorado_bgimg 20220211	no_trump	trump	no_trump	ocr
parson_for_missouri bgimg_20220211	no_trump	trump	no_trump	ocr
re_elect_william fowler_bgimg 20220211	no_trump	trump	no_trump	ocr
senator_tom_james bgimg_20220211	no_trump	trump	no_trump	ocr

Image	OCR Prediction	CNN Prediction	True Class	Which Correct
team_doyle_bgimg 20220211	no_trump	trump	no_trump	ocr
the_boldyga committee_bgimg 20220211	no_trump	trump	no_trump	ocr
van_taylor_bgimg 20220211	no_trump	trump	no_trump	ocr
velez_for_wisconsin bgimg_20220211	no_trump	trump	no_trump	ocr
vicky_l_cook_for fork_supervisor bgimg_20220211	no_trump	trump	no_trump	ocr
vinny_panico_for assembly_bgimg 20220211	no_trump	trump	no_trump	ocr
west_virginia republican_party bgimg_20220211	no_trump	trump	no_trump	ocr
180515-h-ni589-449 extra_20220219	no_trump	trump	trump	cnn
190829-d-bn624-0276 extra_20220219	no_trump	trump	trump	cnn
african_american history_month reception_extra 20220219	no_trump	trump	trump	cnn
barry_moore_bgimg 20220211	no_trump	trump	trump	cnn
billy_long_bgimg 20220211	no_trump	trump	trump	cnn
cbp_tours_san_diego border_wall prototypes_with potus_extra 20220219	no_trump	trump	trump	cnn
donald_trump_and henry_extra 20220219	no_trump	trump	trump	cnn
donald_trump_and mike_extra_20220219	no_trump	trump	trump	cnn

Image	OCR Prediction	CNN Prediction	True Class	Which Correct
donald_trump_and peter_extra 20220219	no_trump	trump	trump	cnn
donald_trump_rally in_youngstown_july extra_20220219	no_trump	trump	trump	cnn
doug_burgum_for north_dakota_bgimg 20220211	no_trump	trump	trump	cnn
elise_stefanik bgimg_20220211	no_trump	trump	trump	cnn
g7biarritz_extra 20220331	no_trump	trump	trump	cnn
gabriel_whitley bgimg_20220211	no_trump	trump	trump	cnn
harmeet_dhillon speaks_at_the_white house-27s_social media_summit_extra 20220219	no_trump	trump	trump	cnn
lagop_gotvr_dec2016 176_extra_20220219	no_trump	trump	trump	cnn
mo_brooks_bgimg 20220211	no_trump	trump	trump	cnn
president_donald_j trump,_senator_tom cotton,_and_senator david_perdue, august_2,_2017 extra_20220219	no_trump	trump	trump	cnn
president_donald_j. trump_delivers remarks_on_5g deployment technology_in_the united_states_extra 20220219	no_trump	trump	trump	cnn
president_trump_and the_indy_500_winner extra_20220331	no_trump	trump	trump	cnn
president_trump announces_1.8 billion_in_opioids grants_extra 20220219	no_trump	trump	trump	cnn
president_trump_at dday75_extra 20220331	no_trump	trump	trump	cnn
president_trump_at the_white_house summit_on_human trafficking_extra 20220219	no_trump	trump	trump	cnn

Image	OCR Prediction	CNN Prediction	True Class	Which Correct
president_trump_in iowa_extra_20220219	no_trump	trump	trump	cnn
president_trump signs_an_executive order_extra 20220331	no_trump	trump	trump	cnn
president_trump welcomes_the_prime minister_of_the slovak_republic_to the_white_house extra_20220219	no_trump	trump	trump	cnn
roast_and_ride extra_20220219	no_trump	trump	trump	cnn
roger_penske receives_the_medal of_freedom_extra 20220331	no_trump	trump	trump	cnn
save_america_bgimg 20220211	no_trump	trump	trump	cnn
steven_palazzo bgimg_20220211	trump	no_trump	trump	ocr
the_2019_prison reform_summit_and first_step_act celebration_extra 20220219	no_trump	no_trump	trump	neither
the_congressional picnic_extra 20220331	no_trump	trump	trump	cnn
the_hispanic heritage_month reception_extra 20220219	no_trump	trump	trump	cnn
the_north_dakota state_bison_visit the_white_house extra_20220331	no_trump	trump	trump	cnn
trump_nashville extra_20220219	no_trump	trump	trump	cnn
trump_welcoming kenya-e2-80-99s president_uhuru extra_20220219	no_trump	trump	trump	cnn
UNGA_extra_20220331	no_trump	trump	trump	cnn
white_house_press briefing_extra 20220219	no_trump	trump	trump	cnn

Raw OCR Results

The OCR characters for the classification disagreements largely look like gibberish, though some valid words are recognizable. I suspect `tesseract` is struggling with images that contain text.

```
texts <- sapply(combined[disagreements, "ocr_output"], `[[`, "word") %>%
  sapply(paste, collapse = " ") %>%
  sapply(strwrap, width = 20) %>%
  lapply(paste, collapse = "\n") %>%
  mapply(FUN = c, sQuote(basename(combined[disagreements, "image"])), .) %>%
  unname()

size <- 10
do.call(what = cat, append(paste(
  rep(c("Image:", "Text:"), times = size),
  texts[seq_len(size)]
), list(sep = "\n\n")))
```

Image: ‘area_25_republican_committee_bgimg_20220211.png’

Text:

Image: ‘austin_chenge_for_governor_bgimg_20220211.jpg’

Text: Tae | r id i Tat) or | 2 y i ’ v0" a FOR GOVERNOR :

Image: ‘brian_for_better_nj_inc_bgimg_20220211.png’

Text: y ace oo ’¥ Me |e

Image: ‘byron_donalds_bgimg_20220211.png’

Text: nn Q ——————————_. §§ = ,—r—“‘é=éeee REPUBLICAN FOR CONGRESS et ay 4 vi) a = — ___ - i) a a

Image: ‘carlos_gimenez_bgimg_20220211.png’

Text: « ad 7 id : ins j 1 qi a) a) b> =, : ae) . Bn - Di.] c: pt » of) Ss = , ’ re R ’ SP . & A) x a ’

Image: ‘area_25_republican_committee_bgimg_20220211.png’

Text:

Image: ‘austin_chenge_for_governor_bgimg_20220211.jpg’

Text: Tae | r id i Tat) or | 2 y i ’ v0" a FOR GOVERNOR :

Image: ‘brian_for_better_nj_inc_bgimg_20220211.png’

Text: y ace oo ’¥ Me |e

Image: ‘byron_donalds_bgimg_20220211.png’

Text: nn Q ——————————_. §§ = ,—r—“‘é=éeee REPUBLICAN FOR CONGRESS et ay 4 vi) a = — ___ - i) a a

Image: ‘carlos_gimenez_bgimg_20220211.png’

Text: « ad 7 id : ins j 1 qi a) a) b> =, : ae) . Bn - Di.] c: pt » of) Ss = , ’ re R ’ SP . & A) x a ’

`tesseract` also returns its level of confidence (a percentage) in each identified word.

```
combined[["average_confidence"]] <-
  sapply(combined[["ocr_output"]], function(x) {
    if (nrow(x) > 0) {
      mean(x[["confidence"]])
    } else {
      NA_real_
    }
  })
```

The average is 59.354.

The mean average confidence is quite different for observations where the classifiers disagreed.

```
t.test(average_confidence ~ disagreements, data = combined)
```

```
Welch Two Sample t-test
```

data: average_confidence by disagreements t = 15, df = 142, p-value <2e-16 alternative hypothesis: true difference in means between group FALSE and group TRUE is not equal to 0 95 percent confidence interval: 26.3 34.3 sample estimates: mean in group FALSE mean in group TRUE 62.5 32.3

Interestingly, it is nor much higher for cases where only the OCR prediction was correct than for those where only the CNN prediction was correct.

```
tapply(combined[["average_confidence"]], combined[["prediction_category"]],
  mean,
  na.rm = TRUE
) |>
  as.list() |>
  list2DF()
```

both	cnn	neither	ocr
62.5	28.3	24.2	36.6

I also check the number of recognizable words `tesseract` identified in each observation by lemmatizing the words and confirming their presence in a reference of English words

```
library(quanteda)
words <- union(tidytext::stop_words[["word"]], tidytext::parts_of_speech[["word"]])
path <- udpipe::udpipe_download_model(language = "english")[["file_model"]]
English <- udpipe_load_model(file = path)

# See https://stackoverflow.com/questions/46731429/quanteda-fastest-way-to-replace-tokens-with-lemma-fr
combined[["valid_words"]] <- sapply(
  combined[["ocr_output"]],
  function(x) {
    if (nrow(x) > 0) {
      tokenized <- tokens(x[["word"]],
        remove_numbers = TRUE, remove_punct = TRUE,
        remove_url = TRUE, remove_symbols = TRUE
      ) %>%
        tokens_tolower() # %>%
      # tokens_wordstem() %>%
      # tokens_replace(pattern = lexicon::hash_lemmas$token, replacement = lexicon::hash_lemmas$lemma)
      raw <- unlist(tokenized)
      if (length(raw) == 0) {
        out <- 0
      } else {
        # Some lemma sequences longer than original word sequence - odd, but likely not important
        lemmas <- udpipe::udpipe_annotate(English,
          unlist(tokenized),
          tagger = "none"
        )[["x"]]
        out <- sum(unname(lemmas) %in% words)
      }
    } else {
      out <- NA
    }
    out
  }
)

# Clean up unneeded file
if (file.exists(path)) invisible(file.remove(path))
```

Most observations seemed to contain at least some valid words.

```
summary(combined[["valid_words"]]) %>%
  as.list() %>%
  list2DF()
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0	1	3	11.2	6	862	24

Overall accuracies are almost identical.

```
colMeans(combined[, c("pred_class", "ocr_pred_class")] == combined[["actual_class"]]) |>
  as.list() |>
  list2DF()
```

pred_class	ocr_pred_class
0.946	0.948

Conclusions

OCR seems capable of accurately reading text in at least some cases, but often outputs gibberish. It could be improved by experimenting with parameters, although there is no guarantee that would prove worthwhile. The convolutional network is surprisingly effective, given the mixed training data. I should create more examples of images with Trump to obtain a more accurate view of performance. While further fine-tuning is clearly necessary, I think the strategy of combining a neural network with OCR remains workable. One issue I did not address is how to combine outputs from both methods into a single prediction.