The *place* (physical and conceptual) of advertising in video game magazines

Hugo Hueber (*Cybersecurity, MSc*) & Florine Réau (*Data Science, MSc*)
Supervised by
Yannick Rochat (*UNIL GameLab, PhD*) & Magalie Vetter (*PSL - École nationale des chartes, MSc*)

Abstract—The video game press has undergone many changes since the 1980s–and the business of advertising has followed. We want to study the evolution of advertising through this specialized press. We use Selective Search to identify areas of interest in a magazine, set up a pre-trained Region-Based Convolutional Networks (R-CNN) then specialized on *Génération 4* magazine to recognize and classify different parts of a magazine, in particular the advertisements. Our model obtains consistent results, and could further be deployed to produce specialized models per magazine and per time period.

I. INTRODUCTION

The first French magazine specialized in video games was published in 1982. After a golden age in the 1990s, a reduction in the magazines supply was observed in the 2000s when media started publishing online, with further diversification seen in 2012[1]. Based on this challenging history, we want to analyze the evolution of advertising in the specialized press, as well as the techniques used to advertise. We want to apply image recognition and machine learning techniques to determine the place - position as well as aim - of advertising in this very press over time. To our knowledge, few research on this particular subject has been carried out[1].

The technical resolution of text classification is an existing field, but nothing specific has been found for the classification of magazine parts[2], particularly techniques for classifying the pages of paper magazines, in order to be able to differentiate between "advertising" and "non-advertising" pages[1]. Parallels with learning on comics or books could however be done.

II. ANALYSIS OF THE AVAILABLE CORPUS AND FOCUS ON SELECTED EXAMPLES

On www.abandonware-magazines.org, there are 12'328 magazines listed[3]. We identified two problems. The first one is that not all magazines are correctly scanned: some of the scans are of low quality, incomplete, or simply missing. The second is that these magazines have different mock-ups, also evolving separately over time. Therefore, the changing

diversity of magazines would be too broad to study. We thus need to focus on a few selected samples.

If we take *Tilt*, *Joystick Hebdo*, *Joystick* and *Génération 4*, over the years 1988-1998, without the special series, we obtain 326 digitized copies. With an average of 120 pages per issue[4], we have about 39'000 pages to study.

III. MODELS AND METHODS

We considered that the detection of magazine pages could be achieved by following a pattern recognition: summary pages always respect a kind of hierarchy, article pages generally have a lot of text, advertising pages often have a main image, etc. So we had to think about crossing the magazines to check these patterns.
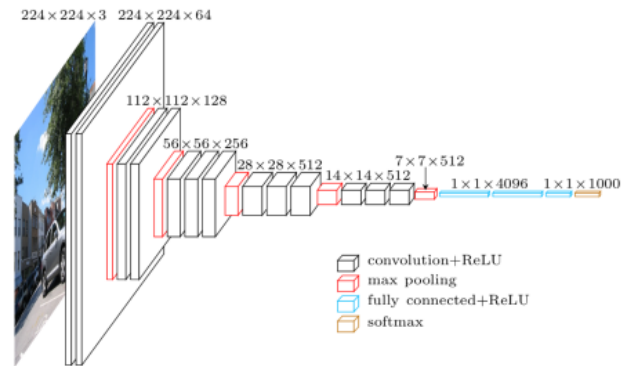
A. Model selection

Figure 1. VGG16 architecture[5]

Our model is based on the Region-based Convolutional Network (R-CNN)[2]. We have chosen this model on the one hand on the advice of our tutors, on the other hand because they have a reputation for being particularly effective in image recognition[6].

Many alternatives and evolutions of R-CNN exist and continue to appear in the domain[3]. We have chosen to stick with R-CNN, because of the apparent ease of implementation. A further work will have to be done to compare the other models.

It is to be considered that learning for image recognition is something that is regularly done. We have thus made the

---

[1]This was discussed with our supervisors. We only have https://www.ludov.ca/fr/documentation/evolution-de-la-publicite and [?]

[2]Some research has been pursued at Montreal, without publication.

[3]https://abandonware-magazines.org/liste.php?tri=identifiant

[4]The selected publications are large editions.

[6]https://www.quora.com/What-is-the-difference-between-CNN-and-RNN

choice to use transfer learning[4], and to reuse existing training. For this particular problem, we went with VGG16[5], and loaded pretrained weights from the ImageNet dataset.

We froze the first layers (namely, pooling and convolution), and only modified the fully connected layers, that we dynamically defined regarding the number of classes. For the loss function, we are using categorical cross-entropy[7], because our model should output probabilities over the set of all classes.

B. Data cleaning and processing

1) "Data concierge": As mentioned above, the data come from the website https://abandonware-magazines.org. The data consist only of scans in image form, without integrated text[8]. The labelling of the images had to be done manually, with the LabelImg[9] software.

Due to a lack of continuity in the data on the one hand[10], and a lack of time to label all magazines on the other hand, we decided to focus on Gen4 magazine. We thus obtained 2'012 pages for our work.

Part of Magalie Vetter's work[11] consisted in selecting the granularity of the classes. The latter were therefore chosen in agreement with her project manager. A future study will make it possible to refine the choice of these classes.

2) Area of interests: For each image, the principle is as follows: first we perform the image analysis to identify areas of interest via selective search[6] implemented in OpenCV. We then iterate on the first N results to classify them with our model, according to a predefined threshold.
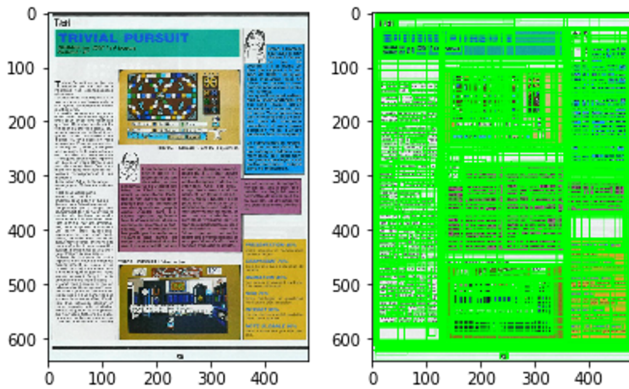

Figure 2. Left: Page example; Right: Area of interests using Selective Search.

<hr/>

[7]https://gombru.github.io/2018/05/23/cross_entropy_loss/

[8]https://en.wikipedia.org/wiki/Optical_character_recognition

[9]https://github.com/tzutalin/labelImg

[10]Some magazines are not complete, some series are not complete, some magazines are not freely available to download, etc.

[11]This project is made for her future Master 2 research thesis entitled "1988-1998: captive press? Presence and impact of advertisers in the French videogame press" (working title). In particular, she wishes to train a neural network to automatically detect the presence and location of advertising in this period's various magazines.

LR (best)	Batch size	Loss	Val loss	Acc	Val acc
0.004	16	7.8226	8.2037	0.5147	0.4910
0.002	32	7.2733	8.3210	0.5488	0.4838
0.003	64	7.5174	8.7426	0.5335	0.4576
0.002	128	8.1850	7.6813	0.4922	0.5234

Hyperparameters comparison with Adam optimizer

LR (best)	Batch size	Loss	Val loss	Acc	Val acc
0.002	16	7.7360	7.5656	0.5128	0.5306
0.002	32	7.3539	7.8979	0.5437	0.5100
0.002	64	7.1956	8.7426	0.5536	0.4576
0.002	128	7.3665	7.4924	0.5430	0.5352

Hyperparameters comparison with Nadam optimizer

A significant amount of these areas are of no interest. However, we can keep some of them to be able to make "counter-examples" and classify the "background" zones.

C. Hyperparameters, preprocessing, and other long words

1) Preprocessing: The data is not annotated on the website, we have to do it by hand. We then receive the annotations as XML files in PASCAL VOC format, for images in the JPEG format. In order to adapt them to the model, we have created two classes, one to process the annotations and convert them to the right format for our model, while saving them to disk for easy access, the other to process the images and build the corresponding extended dataset - again, with the possibility to store them on disk.

2) Hyperparameters tuning: We train multiple models on three parameters, namely the optimizer (Adam or Nadam[12]), the learning rate and the batch size (16, 32, 64, 128, dynamic). The learning rate was tested between 0.001 and 0.01, while the dynamic batch size was basically a function of the samples number. We tried to optimize the accuracy and reduce the loss.

The best way to balance accuracy and speed would thus be to use the Napam optimizer, along with a learning rate of 0.002, and 128 samples per batch. We did not use batch[13] at first, but turns out it boosted the learning process, because less cycles are wasted. A further work could add the use of GPU as a basis.

IV. RESULTS

We trained the model on 100 epochs, with 512 pages. We obtain a final accuracy of about 0.5703. The model seems to work and achieve to recognize images, although it is far from perfect. For example, if we use photos from the ground truth, it is obvious that some labels are missing, others are over-represented. The probability analysis given by our model shows us that the final probabilities are rather low (around 0.30), the use of an adapted threshold allows us to obtain more convincing results.

<hr/>

[12]https://keras.io/optimizers/

[13]https://medium.com/analytics-vidhya/train-keras-model-with-large-dataset-batch-trainin
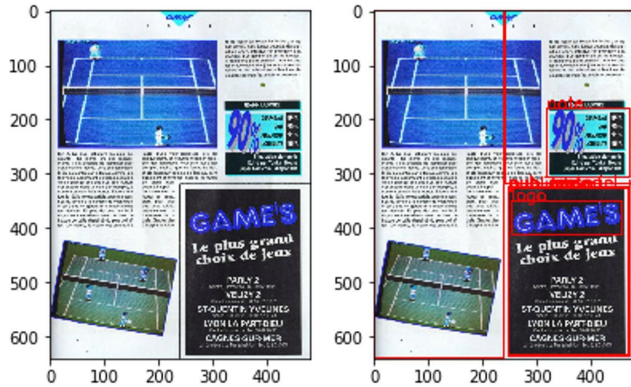
Figure 3.   Left: Page example; Right: Ground truth.

Figure 4.   Model labelization, threshold of 0.85.

When confronted with unknown data, the model responds pretty well. We may want to adapt and specialize it each time we are changing era or magazine.

V. Further researches

The magazines studied are composed of texts and images. In this project, we focus on the differentiation between text and image, one could go further and make a differentiation at the text level (for example between an editorial and an article, or a "normal" article and a sponsored article), and the use of text recognition could be adapted. One could imagine adapting the current pipeline to add a text recognition step, and increase the accuracy of the classification. However, as mentioned above, the text is not integrated into the scan, and pre-processing should also be added in the pipeline. Solutions like Google Vision[14] exist.

Also as indicated above, the granularity of the labels would need to be reworked. One solution could be to make thinner labels ("image", "title", etc.), and assign them a "score". The probability of a page being an advertisement could thus be calculated with these then weighed results.

---

[14]https://cloud.google.com/vision/

References

[1] C. Blandin, *Manuel d'analyse de la presse magazine*. Paris, Armand Colin, 2018, ch. La presse vidéoludique (Björn-Olav Dozo et Boris Krywicki), read November 19th, 2019.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142–158, Jan 2016.

[3] R. Gandhi. (2018, July 9th) R-cnn, fast r-cnn, faster r-cnn, yolo — object detection algorithms. Last visited December 18th, 2019. [Online]. Available: https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365

[4] P. Jay. (2017, April 15th) Transfer learning using keras. Last visited December 17th, 2019. [Online]. Available: https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.

[6] J. Uijlings, K. Sande, T. Gevers, and A. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, pp. 154–171, 09 2013, last read December 17th, 2019.

[7] F. L. (2019) www.abandonware-magazines.org. Last visited Novembere 21th, 2019. [Online]. Available: https://abandonware-magazines.org/

[8] P. Sharma. (2018, October 11th) A step-by-step introduction to the basic object detection algorithms (part 1). Last visited December 1st, 2019. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

[9] ——. (2018, November 4th) A practical implementation of the faster r-cnn algorithm for object detection (part 2 – with python codes). Last visited December 1st, 2019. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/11/implementation-faster-r-cnn-python-object-detection/

[10] ——. (2018, December 6th) A practical guide to object detection using the popular yolo framework – part iii (with python codes). Last visited December 1st, 2019. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/12/practical-guide-object-detection-yolo-framewor-python/

[11] T. Chapuis. (2014, April 11th) 30 ans d'évolution des publicités des jeux vidéos. Last visited December 11st, 2019. [Online]. Available: https://www.konbini.com/fr/inspiration-2/30ans-publicites-jeux-video/

[12] U. Karn. (2016, Augusth 11th) An intuitive explanation of convolutional neural networks. Last visited December 9th, 2019. [Online]. Available: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

[13] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[14] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[16] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[17] (2018, November 20th) Vgg16 – convolutional network for classification and detection. Last visited December 18th, 2019. [Online]. Available: https://neurohive.io/en/popular-networks/vgg16/

[18] "Other references used," we tried to be exhaustive and to not forget any link. [Online]. Available: https://pastebin.com/jBDxGhhH