# ARTIFICIAL INTELLIGENCE IN MEDIA – MAKING SECURITY SMARTER

Milosh Stolikj[1], Dmitri Jarnikov[1,2], Andrew Wajs[1]

1. Irdeto B.V., the Netherlands
2. Eindhoven University of Technology, the Netherlands

## ABSTRACT

Pay TV has evolved from a walled garden, set-top box, model to include online services. Although there are numerous operator and consumer benefits that result from this shift, it also opens up tremendous piracy threats that are a nightmare to control. The sheer volume of information being shared in a more open environment, means that manpower alone is insufficient to process and detect threats effectively. As Artificial Intelligence technology develops in the media space, its application in security must focus on more than closing gaps and locking down assets. Security threats must be spotted and managed faster and more efficiently, before a security instance even occurs.

In this paper, we explain how to leverage Artificial Intelligence to fight piracy by using content monitoring solutions that search and identify pirated content on the internet. At the core of this technology is an AI-powered computer vision system that identifies the original source of distributed content based on the visual information present in the image (e.g. a broadcaster logo). We cover practical issues around building such a system, including its workflow, training and performance.

## INTRODUCTION

Artificial Intelligence (AI) is becoming entrenched in our day-to-day lives. Due to rapid innovation over the last few years, AI is now capable of simulating a range of human brain functions, including pattern recognition. AI is disrupting a variety of industries, including the pay media industry, where it is being considered for implementation of complex security and anti-piracy schemes. In this paper, we present how AI can assist in fighting piracy.

It is widely believed that the most significant threat to the content production and distribution businesses today is the illegal redistribution of content. Pirates recompress content that has been decrypted using a legitimate subscription and stream the content either online or via various streaming IPTV devices. The business model for these types of piracy includes advertising, hardware sales and subscriptions.

A typical example of such illegal activity is near real-time redistribution of live sports events [10]. Such events have high value when they are live. Pirates publish links on social media and aggregation sites to illegal content streams. They re-broadcast the original content with limited modifications (Figure 1). Redistribution is done either via web streaming, with

or without using a Content Delivery Network (CDN), or via Peer-to-Peer applications such as Sopcast or Acestream. This form of piracy has become widespread and it is hard to analyze which streams of content are decoys and which are actual pirated streams.

One of the problems with the metadata that is provided along with the pirated content (if any), is that it is unstructured and inconsistent. This means that it is impossible to determine the actual content in a re-broadcast video stream without analyzing the stream. This is further complicated when multiple television channels are transmitting the same sports event. Even if the event has been identified as being a specific football match, the logo of the broadcaster is the only indication of the original source of the pirated content.

The recognition of the logos in the video content is made even more difficult by the varying qualities of encoding applied to redistributed video. The reproduction quality of logos can be extremely poor, and logos can even be distorted or purposefully hidden.

The media industry and anti-piracy companies have sought to address these issues by applying machine learning (ML). ML can be used to automatically process streams distributed by pirate aggregation sites or other distribution media, and recognize the original source of the video stream by identifying the broadcaster logo.


Figure 1 – Example of captured streams, with distorted logos due to re-broadcasting.

## ROLE OF ARTIFICIAL INTELLIGENCE IN BROADCAST MEDIA

Combating redistribution piracy requires the proactive search and identification of illegal re-broadcasts. AI plays a key role by enabling detection of illegal streams, through semantic analysis of social media advertisements and/or web page indexes and by enabling inspection of visual elements in the re-distributed content, matching it to the original source. Visual elements for identification can vary from the type of content (i.e. football match, movie, new show etc. [11]), the presence of broadcaster logos, detection of known people etc. [12].

Recent advances in ML have shown that neural networks can be designed and trained for various image recognition tasks with high accuracy. Neural networks are attractive because they can learn to distinguish patterns in data by only looking at the input data and the expected output, with no human intervention. CNNs are a sub-class of neural networks which, among other transformations, learn and apply convolutional filters on input data. CNNs are currently state-of-the art for many image processing tasks, significantly outperforming previous methods based on detecting manually crafted features [7].

The major barrier to effectively utilizing a CNN is the training process. This requires a large dataset that encompasses examples of all the possible classification outcomes, as well as

large computational resources to train the networks using the training data. The dataset requires labelling of the data to indicate to which classification group each element of the data belongs to. For example, when training to recognize a specific logo, there needs to be thousands of images of the logo with different levels of degradation, images without the logo and images with other logos.

Razvian *et al* [5] showed that CNNs demonstrate generic feature extraction capabilities when processing images. They showed that the same network architectures can be re-purposed to many different visual recognition tasks, by only changing the training data. Additionally, they showed that a network trained for a task, such as detection of objects [7], can be partially re-trained with little data, for a different task, such as recognizing faces or recognizing alphanumeric characters. We build upon this knowledge to develop a system for recognizing logos.

We have focused on practical issues in using ML to analyze video. We describe the complete workflow of the system for the use case of monitoring and detecting pirated live broadcasts. We show the design choices when selecting the architecture of the ML system for detecting and recognizing logos, which impact the amount and type of data required for training CNNs, and the performance in terms of accuracy.

## USE CASE: LARGE-SCALE MONITORING OF ILLEGAL SPORT RE-BROADCASTING

Monitoring illegal re-broadcasting is a four-step process:

- Discovery of illegal re-broadcasts
- Gathering data from illegal re-broadcasts
- Analyzing data
- Taking measures against re-broadcasters.

The process starts with **discovery** of links to illegal re-broadcasts. Advertisements for illegal re-broadcasts are typically found on social media and indexing websites. Embedded in these advertisements are URIs specifying the protocol used to deliver the web stream (e.g. web stream, P2P stream etc.).

Next, the validity of the stream needs to be verified. This is done by accessing the stream, and **gathering data** from it. The access method differs per protocol. The data consist of the actual content that is being broadcasted, and identifiers of the source of the stream.

The gathered data is then **analyzed**, to identify its original source. The range of tools for analysis can vary from forensic fingerprinting, watermarking to visual content inspection.

Based on the analysis and the information gathered on the source of the stream, the system may act by **taking measures** such as automatic reporting to the re-broadcaster's internet service provider. Further analysis may be performed to identify any forensic marks inserted into the content that enable the tracing of the original subscription used to access the content. The redistribution can be disrupted through terminating access to the service.

Here, we focus on the third step: analysis of visually identifiable information. As identifiers, we use broadcaster logos, which are present in virtually all frames from the original broadcast. Their detection can be done independently, without integration with the content production and delivery system of the content distributors themselves. The system takes a frame from a broadcast stream as an input, and detects whether the image frame has a

broadcaster logo, and then identifies the logo. We expect the system to work in a noisy environment, where images are of poor quality, with distorted and/or partially visible logos.

## LOGO DETECTION AND RECOGNITION

Logo recognition is a common problem in areas such as marketing, brand tracking [1], document classification [2], piracy detection, etc. Depending on the application area, logo recognition can be divided into three sub-problems: detection, localization and recognition.

The goal of logo detection is to decide whether an input sample, such as an image or a video sequence, contains a logo of interest. Logo localization further provides information where the logo is located within the input, most commonly by providing a bounding box around it. Finally, logo recognition or classification deals with deciding which exact logo is found within a sample.

There is an abundance of related work on logo detection and recognition. Early work separates the tasks of describing features and matching features for classification. Feature descriptions can be completely customized [2], or well-known such as Scale Invariant Feature Transform (SIFT) [3,9], and Histogram of Oriented Gradients (HOG) [13]. Feature matching is then treated as an ML problem, with Bag of Words [4], Support Vector Machines or Neural Networks [13] used as classification algorithms.

Newer work on logo detection and recognition relies on CNNs. With these methods, feature description, extraction and matching is learned as a complete pipeline, directly from the training data. These methods yield comparable or higher accuracy than using the previously mentioned feature based methods. However, they require significantly more training data and have higher computational requirements.

Iandola *et al* [8] described how CNNs, pre-trained on a generic image recognition task [7], can be easily fine-tuned for logo detection and recognition, and achieve state-of-the-art accuracy. Similarly, Hoi *et al* [1] compared several CNNs for logo and brand recognition on a new, large-scale dataset. However, at the time of writing, the dataset has not yet been publicly released.

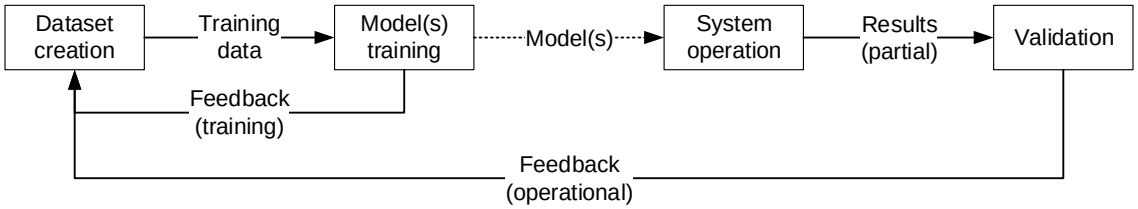


Figure 2 – Workflow for development and integration of ML based system.

## LOGO DETECTION AND RECOGNITION WITH CNNS

Obviously, none of the existing solutions for logo detection and recognition can be directly used since they were not trained to recognize the same logos, nor were they trained to recognize logos in the distorted environment where we find them. Therefore, we must create a dataset for training and select a method for detection and recognition. We also need to continuously adjust the system to accommodate requests for new logos and to improve accuracy of recognition. The workflow for the solution is shown in

Figure 2. In the current paper, we focus on dataset creation step and discuss selection of architecture/models for the solution.

### Dataset creation

Fortunately, there is an abundance of digital material from video broadcasts. We use such material to create a large-scale dataset. We use the dataset to train all CNN architectures described in the previous section, without any pre-training.

Currently, the dataset consists of captured content of 133 different channels from broadcasters that we work with. For each broadcaster stream, we first semi-automatically mark the approximate position where the logo appears. Based on the marked position, we extract the logo and the surrounding area.

To enlarge the dataset, and make training robust, we vary the type of content from which we extract the samples as much as possible (i.e. use different broadcasts from the same channel). Furthermore, we apply data augmentation, by shifting samples in all directions, so that the logo does not always appear centrally, we apply scaling (i.e. zoom effect), and we generate synthetic images by inserting vectorised logos of broadcasters in image samples which initially do not contain logos.

We organize the dataset in two collections. The first collection, named **Logo detection**, consists of two classes of samples: one class with samples of known logos, and a second class of 'noise' data, i.e. samples without a logo. The second collection, named **Logo recognition**, consists of samples of each of the 133 broadcaster logos plus one additional 'noise' class. We found that the addition of the noise class helps with the performance of the models, and enables easier re-training with hard negative examples. We randomly split the two datasets into training and test sets, with sizes as shows in Table 1.

Table 1 – Dataset size for training/evaluation of Logo Detection and Recognition.

| Dataset | Classes | Total number of samples | Number of samples per class | |
|---|---|---|---|---|
| | | | Training | Testing |
| Detection | 2 | 17.810.000 | 6.900.000 | 2.005.000 |
| Recognition | 134 | 8.710.000 | 50.000 | 15.000 |

### Architecture selection

Given the accuracy, flexibility and ease of use of CNNs, they are the preferred method of choice. Still, there is a wide range of options for the selection of an appropriate CNN architecture to use. Currently there is no knowledge of an 'optimal' CNN architecture for logo detection and recognition.

On the one hand, generic CNN architectures have been shown to be applicable for the given task. These architectures tend to be large, with many tuneable parameters, and require significant resources for training and classification. Fortunately, pre-trained models of these CNN architectures are frequently provided. The pre-trained models were trained for generic image recognition tasks, and can be used to bootstrap the re-training process for a new task. The main benefit of using a pre-trained model is that it significantly reduces the amount of training needed to be done. However, this comes with several implications.

First, it does not guarantee that the trained CNN architecture will be optimal for the given task. Second, it fixes the input of the CNN architecture. Typically, all pre-trained networks operate on 224x224 pixel images. Finally, recent attacks on CNN-based recognition systems showed that a certain amount of bias from the training data remains in the trained model [14]. Therefore, an attacker may craft arbitrary images which would trigger the network with relative ease, yielding inaccurate recognitions.

On the other hand, designing and training a customized CNN architecture for logo detection and recognition would most likely produce a much smaller network, optimized for the given task. The amount of resources required to train and use such a network is typically a fraction of the generic state-of-the-art networks mentioned before. However, designing a custom network requires significant investigation in potential architectures, as there are many factors to be investigated. Input size, number and type of layers in the network, activation functions of individual layers, and training methods, are only a few of the factors to consider, which heavily influence network performance.

Considering these design choices, we decided to take a hybrid approach in the design of the logo detection and recognition subsystem. After an initial investigation of different architectures, which is not shown here for brevity, we settled on a mixture of generic and custom networks as a multi-stage ensemble. We designed one custom CNN architecture for logo recognition, and used two other well-known CNN architectures for image recognition, called AlexNet [15] and ResNet [16]. ResNet is currently the state-of-the-art network in image recognition tasks, but is relatively large for the task of logo recognition. AlexNet provides a fine balance between speed of recognition and training, and accuracy.

Table 2 – Characteristics of selected CNN architectures.

| **Architecture** | **Layers** | **Parameters** | **Task** |
|---|---|---|---|
| AlexNet [15] | 8 | 56.8M | Detection, Recognition |
| ResNet [16] | 20 | 14M | Recognition |
| Custom architecture | 8 | 600k | Recognition |

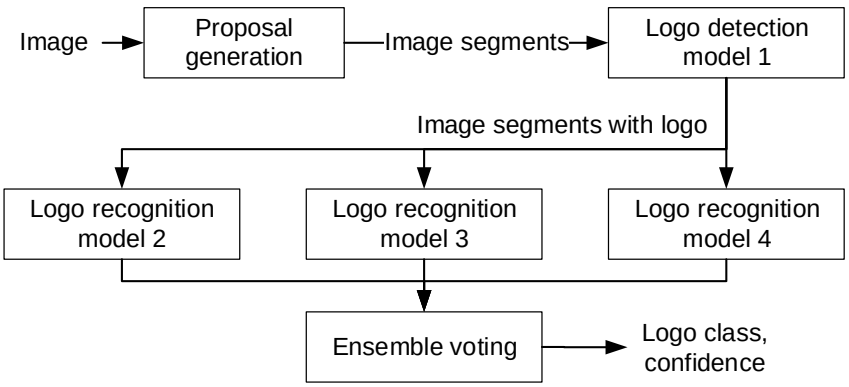


Figure 3 – Architecture of Logo Detection and Recognition subsystem.

First, since current hardware does not allow CNNs to process images of large resolution, we introduce a module for image segmentation, called the proposal generator. This

module segments an image into different pieces at multiple scales. Second, each of the segments is classified using one CNN model which determines whether the segment contains a logo or not. Third, all candidate segments that contain a logo are then classified by 3 different CNN models, with characteristics shown in Table 2. The final classification is done depending on the majority vote of the three models. The architecture of the entire subsystem is shown in Figure 3.

### Results

Table 3 shows the accuracy of the different CNN architectures, trained for the two given tasks. The training was performed using Stochastic Gradient Descent, until the models did not improve accuracy on a randomly selected validation set from the training set. We can observe that precision and recall is similar for both the training and testing set, indicating that the trained models are not overfitting the training data, and are generalizing well. The accuracy of all models is very high, confirming the selection of CNNs for logo recognition.

It is important to note that the best performing CNN architecture, ResNet, is also the slowest one to train and classify. The other two CNN architectures are comparable in terms of processing time.

Table 3 – Accuracy of different CNN architectures for Logo Detection and Recognition.

| Task | Network | Training | | | Testing | | |
|---|---|---|---|---|---|---|---|
| | | Precision | Recall | F1 score | Precision | Recall | F1 score |
| Detection | AlexNet | 99.508% | 99.508% | 99.508% | 99.078% | 99.072% | 99.072% |
| Recognition | AlexNet | 99.585% | 99.585% | 99.585% | 99.480% | 99.480% | 99.480% |
| | ResNet | 99.646% | 99.645% | 99.645% | 99.577% | 99.575% | 99.576% |
| | Custom architecture | 98.544% | 98.512% | 98.523% | 98.438% | 98.407% | 98.418% |

## CONCLUSION

The rapid rise of AI is disrupting many industries. Using AI for automating tasks previously done by human operators allows cost-efficient deployment and development of many large-scale applications that were not feasible before.

In this paper, we presented how AI can contribute to fighting piracy. We analyzed which aspects of AI are applicable in the implementation of large-scale content monitoring systems, which track and identify illegally distributed content. We demonstrated how AI can be trained for automatically recognizing the original source of a video stream, by analyzing the logo of the content distributor from whom the content has been pirated. We see this as a building block into more complex content monitoring systems, which integrate multiple AI components for real-time monitoring of redistribution of copyrighted material.

## REFERENCES

1. S. C. H. Hoi, X. Wu, H. Liu, Y. Wu, H. Wang, H. Xue, and Q. Wu, “Logo-net: Large-scale deep logo detection and brand recognition with deep region-based convolutional networks,” 2015. [Online]. Available: http://arxiv.org/abs/1511.02462

2. G. Zhu and D. Doermann, “Automatic document logo detection,” Conference on Document Analysis and Recognition - Volume 02 (ICDAR). pp. 864–868, 2007.

3. D. G. Lowe, “Object recognition from local scale-invariant features,” Conference on Computer Vision – Volume 2 (ICCV) pp. 1150–, 1999.

4. A. D. Bagdanov, L. Ballan, M. Bertini, and A. Del Bimbo, “Trademark matching and retrieval in sports video databases,” Workshop on Multimedia Information Retrieval (MIR), pp. 79–86, 2007.

5. A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “CNN features off-the-shelf: an astounding baseline for recognition,” IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 512–519, 2014

6. J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?,” Conference on Neural Information Processing Systems (NIPS), pp. 3320–3328, 2014.

7. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” International Journal of Computer Vision (IJCV), vol. 115, no. 3, pp. 211–252, 2015.

8. F. N. Iandola, A. Shen, P. Gao, and K. Keutzer, “Deeplogo: Hitting logo recognition with the deep neural network hammer,” 2015. [Online]. Available: http://arxiv.org/abs/1510.02131

9. R. Boia, C. Florea, L. Florea, and R. Dogaru, “Logo localization and recognition in natural images using homographic class graphs,” Machine Vision and Applications, vol. 27, no. 2, pp. 287–301, 2016.

10. D. Leporini, “Architectures and protocols powering illegal content streaming over the Internet,” in International Broadcasting Convention (IBC), pp. 7, 2015.

11. A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar and L. Fei-Fei, “Large-scale Video Classification with Convolutional Neural Networks,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1725-1732, 2014.

12. O. M. Parkhi, A. Vedaldi and A. Zisserman, “Deep Face Recognition,” British Machine Vision Conference, 2015.

13. K. Li, S. Chen, S. Su, D. Duh, H. Zhang and S. Li, “Logo detection with extendibility and discrimination,” Multimedia Tools and Applications, Volume 72, Issue 2, pp 1285–1310, 2014.

14. S. M. M. Dezfooli, A. Fawzi, O. Fawzi and P. Frossard, “Universal adversarial perturbations,”, 2016. [Online]. Available: http://arxiv.org/abs/1610.08401

15. A. Krizhevsky, I. Sutskever, and G. E. Hinton,“Imagenet classification with deep convolutional neural networks,” Advances in Neural Information Processing Systems 25, pp. 1097–1105, 2012.

16. K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.