POLITEHNICA UNIVERSITY OF BUCHAREST

Doctoral School of Electronics, Telecommunications and Information Technology

Decision No. XXX from DD-MM-YYYY

Ph.D. THESIS SUMMARY

Eng. Mihai DOGARIU

ÎNVĂȚARE ADÂNCĂ PENTRU APLICAȚII CU DATE INSUFICIENTE

DEEP LEARNING FOR SCARCE DATA APPLICATIONS

THESIS COMMITTEE

Prof. Dr. Eng. Mihai CIUC Politehnica Univ. of Bucharest	President
Prof. Dr. Eng. Bogdan IONESCU Politehnica Univ. of Bucharest	Advisor
Prof. Dr. Eng. Constantin PALEOLOGU Politehnica Univ. of Bucharest	Reviewer
Senior Researcher Dr. Eng. Hervé Le Borgne CEA LIST, France	Reviewer
Senior Researcher Dr. Eng. Michael Riegler Arctic University of Norway, Norway	Reviewer

BUCHAREST 2021

This work has been partly funded by the Operational Program Human Capital of the Ministry of European Funds through the Financial Agreement 51675/ 09.07.2019, SMIS code 125125.

Table of contents

Chapter 1

Introduction

1.1 Presentation of the Field of the Doctoral Thesis

Encouraged by the superior performances that deep learning methods have over classical machine learning algorithms, deep learning has become the de-facto approach in the multimedia field. It has benefited from both theoretical and practical research, and this dual aspect can be considered one classification criterion for most works in the multimedia field. Theoretical advances focus on finding the narrowest boundaries on mathematical models that can completely characterize specific tasks, whereas practical research dives into hands-on approaches that require both inspiration and intuition to design new systems which work on real data. These two traits of deep learning are complementary and share a native complexity. It is a field where major breakthroughs on one of the two sides can propel the next breakthrough on the other side.

No matter what algorithm we design or implement, we are required to show its applicability through extensive validation. Traditionally, this has been done by reporting results on openly available datasets which offer the premises for both training and validating algorithms, but they often lag behind the immense diversity that applications have reached. This raises several problems from the datasets gathering process point of view, most of which are related to the actual size of the dataset. It is important to have large enough datasets, preferably consisting of labelled examples and carefully annotated. This type of data is usually difficult to find or gather, adding up to the necessity of having more data available for deep learning applications. We refer to datasets which do not have large amounts of labelled data for precise tasks as being scarce.

1.2 Scope of the Doctoral Thesis

In this thesis we tackle the problem of deep learning for scarce data applications. In this context, we bring a contribution to the practical side of the multimedia field, with

algorithms involving image retrieval and unsupervised data generation for scarce data applications. Therefore, we identify two general ways of dealing with such problems.

The first approach is to design and implement deep learning algorithms that perform well while working with very few examples. The main idea here is to start from models that have very good generalisation capabilities and tune them to very precise applications. This determined us to find hybrid approaches between classical feature engineering and deep learning.

The second idea is to augment scarce datasets with the help of generative models. It is a known fact that gathering a dataset is a tedious process which requires very strict guidelines, significant resources (both time-wise and manpower), and whose outcome might be part of dead research if it is bound too strictly to a given task, without any other applicability for other researchers. Thus, we make use of already existing datasets and extend them by generating synthetic samples that fit the original's dataset characteristics in order to improve the predictive capabilities of models trained on them.

1.3 Content of the Doctoral Thesis

The rest of the thesis contains 3 chapters. The first one is dedicated to image retrieval in diverse applications that work with scarce datasets. It starts with a theoretical introduction concerning image retrieval. The second section presents lifelogging mechanisms that have been deployed in a lifelog moment retrieval benchmarking task over 3 years. In the third section of this chapter we discuss an unattended luggage retrieval system that was designed for Closed Circuit TV systems.

The second chapter addresses the data generation problem. In the chapter's first section we present a theoretical primer on the unsupervised generative models that we used throughout our experiments. Next, we show our progress in the logo generation field. The aim of this part is to extend existing datasets with the scope of improving logo detection performance and add more variability to datasets by using gradient backpropagation. Retrieving the latent code representation that drives a given generation is the key aspect of this section. Lastly, the third section of this chapter focuses on generating financial data under the form of time-series. This part explores a large set of generative models and metrics that can be used to assess the "goodness" of the synthesized samples. As there are no works in the literature covering this specific part for financial time-series, we consider our work to be a leading point for future research.

In the last chapter of the thesis we present a summary of the original contributions and the results that we obtained. We then offer an insight on our future perspectives. Finally, we list the publications that validated this thesis.

Chapter 2

Image Retrieval

This chapter contains the work developed in the context of image retrieval for lifelogging and surveillance applications.

2.1 Theoretical Background

Information retrieval is the process of finding the desired information from a collection of data. The desired information is usually called "search query", or, simply put, "query". Usually, the query does not uniquely identify a single object from the data collection. Instead, it corresponds to a set of such objects, but to different extents of relevancy. Thus, the result comes under the form of a list of items which are ranked according to how well they correspond to the search query's meaning.

2.1.1 Image Meta Search

One way of performing image search is to rely on meta-information such as tags, keywords or key phrases. Each image in the dataset is automatically (or manually) labelled with several concepts, used to create an inverted index system. Then, the query that the user inputs in the search engine is translated into relevant keywords and retrieved from the inverted index ordered by a measure of relevancy.

2.1.2 Content-Based Image Retrieval

An alternative to image meta search is to perform image retrieval based on the content provided by computer vision algorithms. The database here consists of features extracted by computer vision algorithms from each individual image in the collection. During the search process, the same feature vectors are extracted from the query. Then, these feature vectors are compared against the ones stored in the database and the images that bear the highest resemblance to the search query are returned to the user in descending order of their relevancy.

2.1.3 Evaluation

Since image retrieval is expected to return a set of images that respond to a query, it is crucial to have a mechanism which can quantify the performance of such a system. The metrics that are used in the literature to assess how relevant and complete the results of the retrieved results are **precision** and **recall**. Precision represents the ratio between the number of relevant retrieved documents and the total number of retrieved results. Recall represents the ratio between the number of relevant retrieved documents and the total number of relevant results.

Neither of the two previously mentioned metrics is sufficient to completely characterize the retrieval results. This is why these two metrics are usually used in tandem, either expressed as two separate metrics, or combined, under the form of the F-measure: $F_\beta = (1+\beta^2)\frac{P \cdot R}{\beta^2 \cdot P + R}$. Usually, $\beta = 1$ and the F-measure becomes the F_1-measure, representing the harmonic mean between precision and recall: $F_1 = 2 \cdot \frac{P \cdot R}{P + R}$.

2.2 Lifelog Moment Retrieval

Lifelogging is the ensemble of activities through which a person records his/her everyday activities in a digital log. Most of the time, this information is unlabelled, which makes it difficult to organize and navigate making the retrieval of specific events a complex task. The ImageCLEF evaluation campaign proposed to alleviate this issues by introducing the Lifelog Moment Retrieval task (LMR). The following subsections cover our progress during the 2017, 2018, and 2019 LMR competitions.

2.2.1 Literature Overview

LMR is a fairly new concept which has been encountered primarily at benchmarking evaluation campaigns such as ImageCLEF and NTCIR. We briefly present the most relevant efforts in these competitions. One approach involves a pipeline that performs segmentation of the evaluation dataset based on manual introduction of timestamps and metadata consisting of concepts. A similar approach involves running each image through object detection, person counting, and places detection, after which a similarity measure on the feature vector is applied. Our approach [5] made use of a similarity distance based on WordNet. An approach where objects, scene features, and actions were extracted and combined along with textual descriptions in an inverted index was also proposed. Our approaches [6, 7] propose a similar technique, applying a blur threshold system as a primary step, then extracting several features such as places, concepts, objects, and combining them with textual information.

2.2.2 Proposed Approach

A. Datasets Having an accurate description of lifelogging activities implies capturing as many sensory aspects as possible, such as movement, location, actions, etc. We reckon that it is critical to have a good understanding of the available lifelog data in order to perform accurate retrieval. One strong point of the ImageCLEF LMR competition is that the organizers released a new, more diverse dataset each year. We summarize in Table 2.1 the main aspects of the datasets that were used for developing lifelog retrieval algorithms.

Table 2.1 Summary of LMR datasets that have been used from 2017 to 2019. "yes"/"no" marks the presence/absence of the respective feature from the development dataset.

	LMR_2017	LMR_2018	LMR_2019
users	3	1	2
images	88k	80k	81k
image concepts	ImageNet	Microsoft	Microsoft
locations	130	135	61
activity	yes	yes	yes
biometrics	no	yes	yes
music	no	yes	yes
places	no	no	yes
objects	no	no	yes
dev set topics	5	10	10
test set topics	10	10	10

B. Lifelog Moment Retrieval at ImageCLEF 2017 Our work [5] during the ImageCLEF 2017 Lifelog Moment Retrieval Task follows the diagram presented in Fig. 2.3.

Each image from the competition's dataset is assigned a set of 1000 concepts with different confidences. We keep only the most relevant (highest confidence) concepts for each image. Then, we propose an approach that uses WordNet's embedded tools and

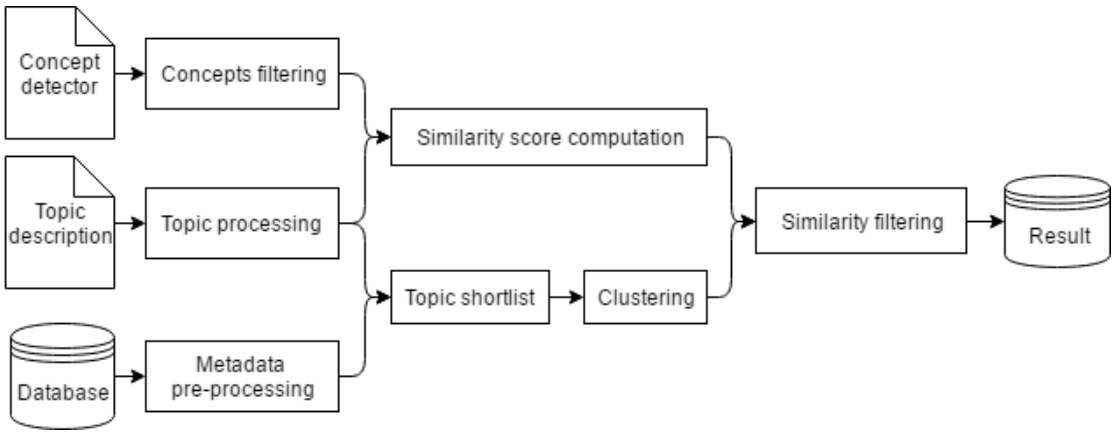

Fig. 2.3 LMR 2017 processing pipeline [5].

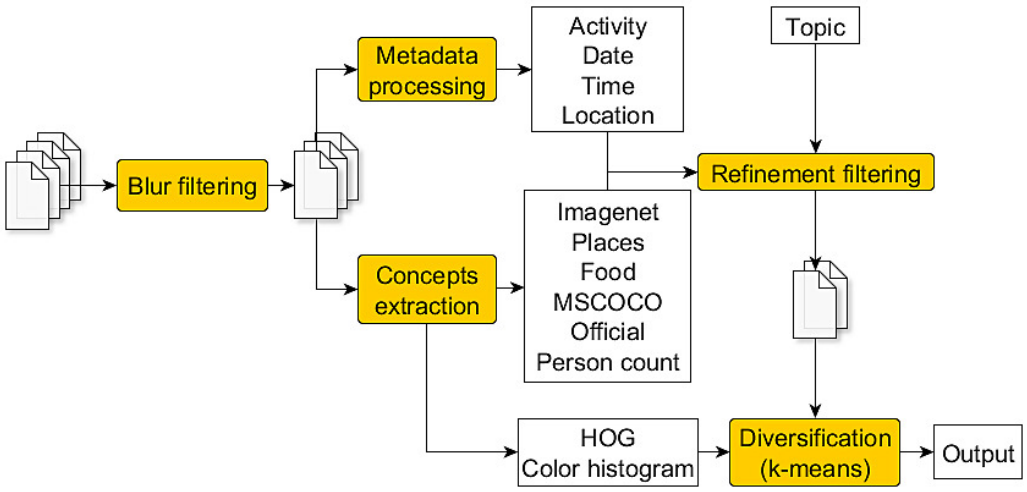

Fig. 2.8 LMR 2018 processing pipeline [6].

keep only the nouns and verbs from the textual description which constitutes our search query. In the end, for each topic we obtain some coarse, but fundamental information related to that specific query. We then discarded images that are in dissonance with the query's metadata (user number, location, activity) drastically reducing the size of the relevant dataset. Next, we implemented a hierarchical clustering algorithm based on the Histogram of Oriented Gradients (HOG) extracted from each image and stopped the hierarchical clustering algorithm when 30 clusters were formed. The similarity measure that we proposed was computed between a set of concepts and a set of words corresponding to a certain topic description. For this, we used WordNet's Wu-Palmer similarity measure. Each distance is weighted with the concept's confidence and we add all distances between (*detected concept*, *description word*) pairs. Finally, once we obtain the clusters and the similarity scores between each image from the cluster and the topic description, we select the best candidates for submitting a run. We sort the clusters in descending order based on the mean value of the similarity scores of the images that it contained. Then, we selected the 2 best ranked images from 25 clusters. The official metric for our run was **F1@10=0.132**.

C. Lifelog Moment Retrieval at ImageCLEF 2018 In this subsection we present the algorithm with which we participated at the ImageCLEF 2018 benchmarking campaign [6]. The pipeline of our system is presented in Fig. 2.8.

We first remove blurry images that have a focus measure below an imposed threshold, Next, we run each images through several classifiers and a detector: Imagenet, Places365, Food101 classifiers and a Faster R-CNN [24] object detector. The object detector's "person" class was also used for person counting. Additionally, we used the official concepts, date, time, activity and location metadata associated to each image, provided by the organizers. We implement two types of refinement filtering. First, we manually interpret

Table 2.3 LMR 2018 official results for the submitted runs.

Run	$F1@5$	$F1@10$	$F1@20$	$F1@30$	$F1@40$	$F1@50$
Run 1	**0.235**	**0.216**	**0.224**	**0.218**	0.203	0.199
Run 2	0.154	0.169	0.215	0.21	**0.207**	0.199
Run 3	0.158	0.168	0.217	0.214	0.199	**0.206**
Run 4	0.129	0.166	0.184	0.184	0.178	0.188
Run 5	0.412	0.443	0.446	0.438	0.419	0.405

the entire topic text and extract meaningful constraints on the metadata associated with each image and remove those that do not satisfy the given constraints. In a similar manner, we remove images which contain certain objects/concepts. We then compute the relevance score as a weighted sum between all the detected concept confidences and corresponding reference vectors. Each image is then represented by the concatenation of two normalized vectors: a 1536-D vector representing the Histogram of Oriented Gradients (HOG) feature vector and a 512-D vector representing the color histogram feature vector. We run the K-means algorithm with either 5, 10, 25 or 50 clusters. For the final list of proposed images we select from each cluster the image with the highest relevance score in a round-robin manner.

We have submitted one run during the competition and 4 other runs after the competition ended. In Table 2.3 we present the final $F1@X$ results that we have obtained for each run with best values in bold. Our last run obtained the best results because it implied a highly supervised approach.**Run 1** follows the pipeline described above, for **Run 2** we excluded newly added images that are too visually similar to the ones already in the list, for **Run 3** we built the reference vectors with the same technique that we used in [5], **Run 4** run was similar to **Run 3**, with the only difference being that all the weights were set to 1. Lastly, **Run 5** was carried out with the same approach as **Run 1**, this time performing a fine-tuning of all system parameters for the topics that had bad results in the first run by trial and error. The official result that we obtained was that of the first run, **F1@10=0.216**.

D. Lifelog Moment Retrieval at ImageCLEF 2019 For the 2019 edition of the Lifelog Moment Retrieval Task our work [7] focused on excluding uninformative images as a first step, and then compute a relevance score for the remaining subset of images. From previous experience, we noticed that being more strict with the criteria for excluding uninformative images leads to better results. The architecture of our system can be seen in Fig. 2.10.

We start our pipeline by running a blur detection system, computing the variance of the Laplacian kernel for each image. The images that do not meet a certain threshold are removed from the pipeline. Next, the metadata of the images is checked to be in accordance with the restrictions imposed by the queried topic. Information about the

user's id number, location, time, action, time zone are then used to remove another part of the remaining images. In some cases, this selection of metadata can suffer modifications from one topic to another. At this point, the set of images has drastically diminished in comparison to the original dataset. Then, we run the remaining set of images through the relevancy score computation process.

The development dataset contained information regarding the detected attributes, categories, and concepts in each image. The attributes refer to different aspects of the scenery, objects and image classification. We also kept track of the number of detections of each object in every image, since each object could trigger multiple detections in the same image. We compute the relevance score as the sum of the confidences of features that need to be detected in the image. As opposed to our 2018 approach [6], we do not use a weighted sum because the weights have to be manually tuned for each individual query and it would stray too much from the idea of automatic processing. In the end, we submitted the 50 best ranked images per topic, according to the relevance score.

We note that the user has to manually select the parameters for both the metadata restrictions and the list of items that drive the relevance score, and this is the only manual input required from the user. As far as we know, there has yet to be developed a clear method on how this parameter tuning process could be completely automatized. The official result of our run was **F1@10=0.127**.

2.2.3 Conclusions

We participated in 3 consecutive editions of the ImageCLEF benchmarking campaign and proposed 3 different approaches to solve the Lifelog Moment Retrieval tasks. In Table 2.8 we present the official results of our systems through the years, along with the best score and the baseline and draw several conclusions. First, it is extremely difficult to design a system that answers retrieval queries with unlimited degrees of freedom. Secondly, truly automatic runs will fall behind those attempts that involve human interfering. Lastly, since the lifelogs cover a small time span, most events are

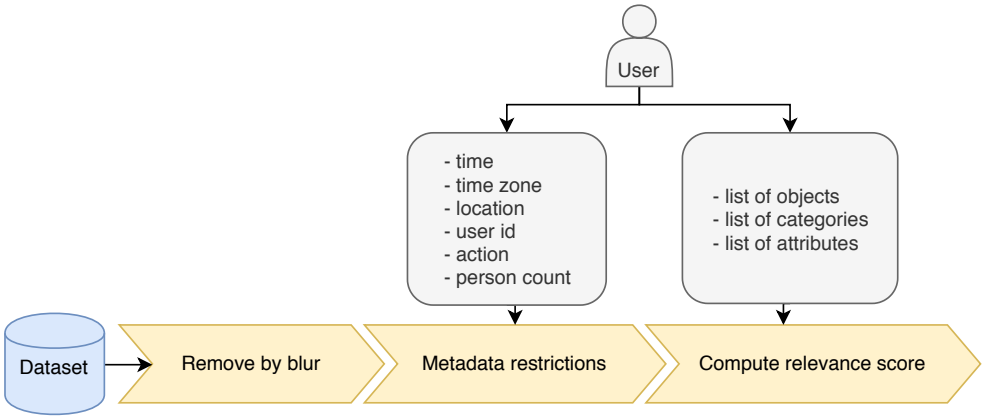

Fig. 2.10 LMR 2019 processing pipeline [7].

bound to occur only a very limited number of times (possibly only once), transforming this task into a scarce data manipulation problem.

Table 2.8 LMR $F1@10$ results from 2017 to 2019. Comparison between the baseline, our best run, and the competition's best run.

Year	2017	2018	2019
Baseline	0.103	0.131	0.289
Our result	0.132	0.216	0.127
Best result	0.497	0.545	0.61

2.3 Unattended Luggage Retrieval

This section covers another type of image retrieval that this thesis addresses, namely unattended luggage retrieval. This problem became more important with the increasing number of explosives that can be concealed in ordinary packages such as backpacks, suitcases or bags and can be placed in crowded places without attracting much attention. Since the ratio of images where unattended luggage are present to the total number of images captured by a surveillance system is close to zero, we are facing a scarce data problem.

2.3.1 Literature Overview

Most approaches in the abandoned luggage detection literature focus on semantically separating the background from the foreground, and then tracking both static and moving objects. Unlike them, we propose a system composed of three modules: an object detection component, a suspect detection subsystem, and a person re-identification component [9]. We use Mask R-CNN [14] to perform object detection and extract all the relevant features. A similar work to ours was conducted by Intel [2], but their approach is limited only to object detection.

2.3.2 Proposed Approach

Our system uses the Mask R-CNN architecture, trained on MS-COCO dataset. From the object classes we select only those that impact our system directly and discard the rest. We will now describe the setup that we used for each of the 3 modules, and how they interact.

Unattended Baggage Detection The unattended baggage detection algorithm makes use of Mask R-CNN's object detection mechanism. We aim to detect only a subset of classes: "person", "backpack", "handbag", and "suitcase". The last 3 classes have been

grouped under a general class regarded as baggage. We labelled an object as unattended when the object's bounding box does not intersect any detected person's bounding box.

Suspect Detection We then use the feature vector that is generated by Mask R-CNN's Region Proposal Network (RPN) and search for the exact abandoned baggage through all of the available images, and rank these images in descending order of feature vector similarity, under the condition that there exists in the image a person whose bounding box intersects the baggage's bounding box. We test for similarity by using a simple Euclidean distance. Afterwards, we run a ranking of these distances and display the images where it is most likely that the abandoned baggage was detected in the presence of a person and deem this person to be a suspect.

Suspect Re-identification In the next step, our system starts from the suspect that was just detected and searches for him in the camera feeds. We used the same procedure to search for the person as we used for the baggage. This time, however, we performed a per-camera ranking, and obtain for each camera a set of images where the suspect was detected. This is motivated by the fact that we want to track this person's path throughout the surveilled perimeter.

Dataset Additionally, we created a small dataset for demonstration purposes. We gathered 1 hour of images recorded by our research center's CCTV system. We restricted the observed area to the basement, ground floor, and the exterior of the building. We established a scenario where one person would abandon a backpack on the hallway and leave. Several other people carrying backpacks were captured in this dataset. In addition, we created a demo graphical user interface to aid a human operator.

Validation and Results We performed tests on several object detection backbone architectures and report average precision for Intersection over Union (IoU) score higher than 0.75[1] along with the time it takes for each architecture to process one image in Table 2.9. The backbone architectures should be read as follows: R/X (ResNet or ResNeXt), 50/101 (number of layers), C4/DC5/FPN (ResNet conv4 backbone, ResNet conv5 backbone or ResNet + FPN backbone, respectively), 1x/3x (number of epochs multiplier).

We obtained the presented inference times while performing the detection on a single NVIDIA QUADRO M4000 GPU. We consider that in the given circumstances it is better to opt for a model which sacrifices a part of the detection accuracy in favor of a faster inference time. The detection accuracy loss can be overcome by setting a lower detection threshold to force additional proposals and decrease the false negative rate. Decreasing

[1]Detection performance results taken from Facebook Research Object Detection MSCOCO baseline: https://github.com/facebookresearch/detectron2/blob/master/MODEL_ZOO.md

Table 2.9 Performance of different object detection models.

Backbone	Bbox AP@IoU=0.75	Inference time (s/image)
R50-C4_1x	35.7	0.392
R50-DC5_1x	37.3	0.408
R50-FPN_1x	37.9	0.228
R50-C4_3x	38.4	0.398
R50-DC5_3x	39.0	0.396
R50-FPN_3x	**40.2**	**0.231**
R101-C4_3x	41.1	0.482
R101-DC5_3x	40.6	0.474
R101-FPN_3x	42.0	0.308
X101-FPN_3x	43.0	0.591

inference time is, however, far more difficult. In our use-case a fast response is a critical aspect of the system. Therefore, we select the R50-FPN_3x as our go-to model in the proposed system.

The person re-identification component was tested on the CUHK03 dataset and obtained a top-1 accuracy of 70.8%. The same technique was used by Xiao et al [29]. During our demonstration we managed to capture all events that are of interest: detect the abandoned baggage, the person that left it there, and then detect that person's presence on individual cameras. Furthermore, we managed to extract important moments, such as when the person entered the building while carrying on the backpack and when the person left the building, without the backpack.

2.3.3 Conclusions

We presented an unattended object detector that can be deployed on CCTV systems. Our approach is composed of 3 modules, each designed to perform a different task but from the same feature vector: unattended object detection, detect the object's owner, and find that person's presence on the CCTV cameras. We used Mask R-CNN to perform the detection and computed similarities on the feature vectors that the network's RPN module extracted. We gathered a small dataset for our experiment simulating a real use-case, built a user interface for operators, and proved that the system works as intended in the proposed scenario.

Chapter 3

Unsupervised Data Generation

This chapter contains the work that was done with the aim of augmenting scarce datasets by using unsupervised deep learning techniques. It begins with a short review on existing unsupervised generative algorithms. Then, we present our efforts in the image generation field by presenting a logo generation application. Next, we focus on time-series generation for stock market analysis.

3.1 Theoretical Background

Internet accessibility is growing at an exponential rate so it is expected that more and more of the new multimedia content will be unlabelled since it is virtually impossible to label data at the same pace as it is being created. Consequently, understanding how to deal with and learn from unlabelled data has become far more interesting than in previous years. There are several popular types of models benefiting from unlabelled data that we used in our work, which we will briefly present next.

3.1.1 Autoencoders

Autoencoders [20] are neural networks that are trained to replicate the input at their output. They are composed of 2 cascaded subsystems, an encoder and a decoder, connected in a feed-forward network. The input x is encoded into the latent vector h, which, in turn, is decoded into the output x'. One downside of autoencoders is that they cannot be used for sample generation. Autoencoders learn the latent code representation of their training data only, not being able to generalize well. This latent code space is almost every time discrete, making the autoencoder unreliable for generation purposes.

3.1.2 Variational Autoencoders

Variational Autoencoders (VAE) [19] are probabilistic directed generative models. Similar to autoencoders, they also employ an encoder, a decoder, and a latent vector for-

mulation, but their training procedure is very different. In the VAE case the generation process starts with a sample z drawn from the latent code distribution $p_{model}(z)$. This is passed through the generator, $g(z)$. Lastly, x is sampled from the distribution $p_{model}(x; g(z)) = p_{model}(x|z)$. In the training step, z is obtained through the encoder network $q(z|x)$. Conversely, $p_{model}(x|z)$ is considered the decoder network. In practice, the randomness of the samples drawn from $p_(z)$ makes it impossible for optimization algorithms to perform backpropagation, so a reparametrization trick is used instead.

3.1.3 Generative Moment Matching Networks

The key idea of GMMNs [21] is the use of a statistical hypothesis testing framework called maximum mean discrepancy (MMD). First, an autoencoder is trained on a given dataset. Next, the encoder is used to transform the input data into the latent code space, forming the discrete latent code distribution. The generator is then trained to sample data from the latent code distribution which, in turn, will be transformed by the autoencoder's decoder into new samples. As previously mentioned, autoencoders have a discrete latent code distribution, which makes them unusable for generation. GMMNs, however, learn a continuos data distribution over the latent code space.

3.1.4 Generative Adversarial Networks

Generative adversarial networks (GANs) [13] are the last type of generative models that have been approached in this thesis. They rely on a game theory scenario where there are 2 parties (generator and discriminator) involved that compete against each other. The adversarial framework can be formulated as a zero-sum game where a function $v(\theta^{(g)}, \theta^{(d)})$ determines the discriminator's payoff. Oppositely, the generator receives $-v(\theta^{(g)}, \theta^{(d)})$ as payoff. With each of the two parties trying to maximize their own payoff, convergence is attained at $g^* = \arg\min_g \max_d v(g,d)$ [12]. At that point, the discriminator will be unable to tell the difference between the real samples and the ones synthesized by the generator and will output $\frac{1}{2}$ everywhere. Now, the training is considered to be complete and only the generator is further used.

3.2 Logo Generation

One application of unsupervised generation that we approached is logo generation. This implies the creation of new company logos with the aim of augmenting already existing datasets. The main application regarding logos is automatic detection, motivated by the fact that companies want to know how visible their product is in different media. Next, we will present our progress [8] in the logo generation field.

3.2.1 Literature Overview

Logo detection is, in fact, an application of object detection, which is a field that reached its maturity. Two-stage detectors [14, 24] have traditionally obtained more accurate results, but at the expense of more complex algorithms that require larger computation time, as opposed to their single-stage counterparts. Another important aspect of logo detection is the dataset that is used to train the detector. Logo detection, as opposed to object detection, is poorly represented, with only a few datasets available for detection, such as FlickrLogos-47 [25]. Additionally, there is the Large Logo Dataset (LLD) [26], a dataset consisting only of digitally represented logos and not in-the-wild instances.

3.2.2 Proposed Approach

Our work aims to alleviate the data scarcity problem by augmenting already existing datasets with synthetically generated logos. In order to generate specific logos our work is structured in 3 steps: 1. train a logo generation model, 2. retrieve the latent vector for a given logo through gradient backpropagation and slightly alter it such that we obtain different instances of a given sample, used for augmentation and 3. validate the approach with a logo detection system.

Logo synthesis The logo synthesis part is done with the help of a generative adversarial network, namely the DCGAN [23] model, trained on LLD. We trained several DCGAN models on LLD under different parameter setups. Since assessing the quality of the generated images is known to be difficult, we manually examined the outputs of all our models in order to decide on the best setup.

Gradient backpropagation GANs transform latent vectors z into images I through their generators, $I = G(z)$. We are interested in the backwards process, namely finding an approximation of the latent vector $z' \approx z = G^{-1}(I)$ that can be used to generate a given logo. The reverse mapping, from images to vectors, can be done by backpropagating the gradients of the cost functions [22]. Given a noise vector of size 100, uniformly distributed between -1 and 1, $z \sim U([-1,1]^{100})$, the generator of a GAN will produce an image $G(z)$. We generate another random noise vector z' and its corresponding image $G(z')$. Then, we want to force z' to be as close as possible to z in order to obtain $G(z')$ as close as possible to $G(z)$. This is done by minimizing the L_2 norm. Since there is also an additional constraint that the noise vector should fall inside the $[-1,1]^{100}$ hyper-cube, a modified optimization is proposed, where each value that falls outside this interval will be replaced with a random value sampled uniformly from the $[-1,1]$ interval.

Logo detection As it was mentioned before, assessing the quality of the samples generated by a GAN is a complicated subject itself. In order to determine how good the

reconstruction is, we setup a logo detection pipeline. Logo detection is, in essence, an object detection problem. Therefore, we applied a Faster R-CNN [24] architecture and adapted it to logos. Each logo from the Flickr_47 dataset was extracted, reconstructed with the backpropagation method and inserted back in the original image. This dataset was added to the already existing training dataset, thus obtaining two versions for each image in the training dataset: one with the original logos and one with the reconstructed logos. We further refer to this dataset as and extended version of Flickr_47. We pre-trained the logo detection algorithm on the MS-COCO dataset, then we fine-tuned it on the extended Flickr_47 train dataset and tested it on the Flickr_47 test dataset. We obtained $\mathbf{mAP@0.5 = 0.6019}$ which is very promising. We also trained another vanilla logo detection model on Flickr_47 training dataset without any reconstructed images and tested it on the Flickr_47 test dataset and obtained $\mathbf{mAP@0.5 = 0.6725}$.

3.2.3 Conclusions

In this section we presented a logo generation algorithm based on GANs and gradient backpropagation. We trained a DCGAN model to generate random logos starting from the LLD dataset. Then, we extracted in-the-wild logos from the Flickr_47 dataset. These logos were backpropagated through the DCGAN, we extracted their latent code representations, and then used these noise vectors to guide the DCGAN's generator to output similar looking logos. These new logos were used to replace their original counterparts and we ran a logo detection pipeline to validate our processing.

3.3 Financial Time Series Generation

Another issue related to data labelling concerns the availability of less common datasets. In this section, we present our exploratory work [1] in the field of automatic generation of realistic financial time series. To capture the underlying properties of the data, we exploit various designs of unsupervised generative models. We introduce the problem of assessing the quality of the artificially generated data and propose several solutions. We also propose a method of generating arbitrarily long financial time series.

3.3.1 Literature Overview

There are two main directions that researchers are following for financial time-series. One approach is to focus on predicting the next sample(s) in a sequence based on the available recent history, similar to a regression problem. The other approach is to generate a fixed number of consecutive samples at a time, in a dataset augmentation manner. This procedure requires to extract windows of a given length L_w when creating

[1] Research has been funded by Hana TI [4, 10]

the training dataset for the GAN. Then, the synthesized samples are usually validated by a stock market prediction algorithm. It also represents the main application for our financial time series generation algorithm.

3.3.2 Financial Data

A company's financial time series represents the chronological evolution of several indicators, denoted OHLC (Open, High, Low, Close). In order to capture the relative difference between consecutive days, log returns will be used:

$$r_i = \log \frac{C_i}{C_{i-1}}, \tag{3.7}$$

where C_i represents the closing price of day i and r_i the log return closing price of day i. This reduces not only the intra-variation of the time series, but also the inter-variation between different companies.

Financial time series are known to be more peaked than normal distributions and exhibit a fat-tailed behavior. Also, large changes of prices tend to cluster together. Lastly, empirical asset returns are uncorrelated for any value of the lag larger than one, but not independent. Generative models should deal in particular with these aspects.

3.3.3 Proposed Approach

This section presents our efforts [4, 10] of generating realistic synthetic financial time series of arbitrary length. We compare our results with the ones of Takahashi et al. [27] since the principle they are following is the closest to our work.

Generative models The first step of our proposed solution consists of generating a fixed-length 1-dimensional array of samples with the help of several generative models. We performed an in-depth study over a vast number of generative architectures. We investigated 3 major classes of generative models: GANs [13], VAEs [19], and GMMNs [21].

Our aim is to generate 1-dimensional arrays of length n and then combine them in arbitrarily long time-series. In the following, we briefly describe the architectures that were implemented. The length of the synthesized 1D array has been set to 250 for all models, the equivalent of an entire working year in finance. Please note that all architectures are presented in their optimized versions, achieved after in-depth ablation studies. We proposed 4 different multilayer perceptrons (MLP_1 to MLP_4), 6 fully convolutional GANs ($FCGAN_1$ to $FCGAN_4$, *snFCGAN* and *FCGANmc*), 2 VAE models and 2 GMMN architectures.

Dataset Creation Our system is designed to generate synthetic financial time series of fixed length. Therefore, we train the generator to output fixed length 1-dimensional arrays or, in the case of the multichannel architectures, 4-dimensional arrays, of size L_w. We train our generative models on the S&P dataset provided by Hana Institute of Technology. This dataset consists of 1,506 companies with daily OHLC records from January 1st 2000 to March 31st 2020. The financial time series start at different times, but end at the same date. We split each available time series into segments of a fixed number of samples using a sliding window. Starting from the earliest position we begin extracting segments of 250 samples for each company. We chose to drop incomplete segments (due to the company being listed on the market during the window's time span) completely. We process the rest of the dataset by sliding a window, with a rolling step of 30 samples, equivalent to 6 working weeks, and add them to our training data set. We denote the obtained training dataset as $D = \bigcup_{\substack{i=1 \\ \text{step } 30}} W_i$, where W_i is the set of all complete segments starting at day i. All data was transformed to log returns, as explained in Eq. 3.7. For the multichannel architectures, we generated Open, High, Low, Close by using four channels instead of one in the fully convolutional setup. We encode the relation between these values and the closing price instead of using their absolute values. Thus, we use $\frac{High_n - Close_n}{Close_{n-1}}$, $\frac{Close_n - Low_n}{Close_{n-1}}$, and $\frac{Close_n - Open_n}{Close_{n-1}}$.

Dataset Preparation Our final dataset, D, contains more than 200k 1-dimensional entries each of length L_w. We propose 2 ways of sampling data from this dataset. First, we consider D to be a homogeneous mixture of windows and randomly sample batches of data from it. We denote this setup $Data_1$. We consider this to be part of a preliminary study that we used to guide our more in-depth research. The second approach is to keep each set of segments W_i in their original form and treat each such subset as an entire batch in our training process. This method feeds the model with cross-correlated data, helping it to inherently learn this property. We denote this setup $Data_2$.

Regime Splits Stock markets are generally "growing", meaning that the values of the closing prices are increasing on the long term. In log return terms it means that the positive values outweigh the negative ones. If the difference between the cross-sectional mean and the cross-sectional rolling mean is positive, then the regime is considered to be up-trending. If it is negative, then the regime is considered to be down-trending.

Performing this split based on regimes results in labelling 68.07% of the days as being up-trending and the rest of 31.93% as down-trending. We decided to follow 2 strategies for training each of our models. First, we trained each model with the entire dataset D. Second, we computed the regimes for each day on the original dataset, split the dataset according to the two regimes, and then applied the windowing mechanism on each of the two regimes. This means that for each architecture setup we will obtain 3 models: a complete one, trained on the entirety of the dataset without regime splits, an

up-trending one, trained only on up-trending days, and a down-trending one, trained only on down-trending days. We denote these 3 versions as "complete', "up', and "down', respectively.

Synthetic Time Series Formation Once we have the generators trained to output a fixed length time series, we need to combine them such that we obtain arbitrary size time series. In the case of the "complete' models, we simply generate several batches of fixed-length and concatenate them together. For the "mixed" regimes approach, however, we apply the following procedure. We rely on the fact that the up-trending and down-trending regimes come in bursts of 20 to 120 (statistically determined) consecutive samples. Moreover, we know the final quota of each of the two regimes. Therefore, we sample chunks of random length between 20 and 120 samples with a 68% probability of them coming from the "up" model generator and 32% of them coming from the "down" model. We concatenate these chunks until we reach the desired time series length. Also, adjusting the batch size for the generator is equivalent to setting the number of stocks that we want to generate for a given period, i.e., the financial universe size.

Evaluation The evaluation of artificially generated data is still an open issue as there are no standardized methods of assessing how well a GAN performs on financial time series. We therefore analyze and propose a series of metrics that were inspired by multimodal signal generation problems. Most of these metrics are still experimental regarding how accurate they can describe the performance of a financial data GAN, but they can still help establish an hierarchy between different models. We approach the evaluation at three levels: (i) qualitatively, (ii) quantitatively, and by (iii) predictive accuracy test, as presented in the following.

1. **Qualitative analysis** — first 4 central moments, autocorrelation, heavy-tailed distribution, volatility clustering, t-SNE for 2D visualization, cumulative sum of the returns, and trend ratios. We manually investigated whether the real and synthetic samples share the same behaviour for all of the above metrics.

2. **Quantitative analysis** — Kullback-Leibler divergence, Jensen-Shanon divergence, Kolmogorov-Smirnov test, and Earth Mover's distance. Small values of the aforementioned measures indicate similar distributions. We chose a random batch of real samples as reference where one was needed.

3. **Predictive accuracy test** — To assess the quality of the synthetic data, we propose to predict stock market movement by framing the prediction as a classification problem, discerning whether stocks are outperforming or underperforming at a predefined time point. Our proposed method consists of four stages: (i) statistical clustering of stocks based on their normalized returns, (ii) statistical labelling

of stocks in outperformer or underperformer, (iii) denoising and dimensionality reduction, and (iv) training predictive models to generate the one-step-ahead output. In the prediction stage, we run our simulations over 20 years (going back from 2000 up to 2020), using a two-part split protocol. In the training part, we use the past seven years' worth of data to train the models using a mix of real and synthetic data, and in the testing part we use the following next year using strictly only real data to predict performers and underperformers.

Results and discussion

We made snapshots of each network setting whenever it would encounter a new best value for any of the proposed quantitative metrics. Afterwards, we went through all of these snapshots and manually inspected all the qualitative metrics. Empiric results show that among the proposed metrics, the Jensen-Shannon divergence is the best indicator to which model has a better overall performance, so we present the results for the snapshots that achieved the best JSD for each network model.

We run a preliminary experimental stage on the data from the $Data_1$ dataset implementation. Then, we train the models presented in Table 3.3 (marked with the [4] citation) and use them to generate only fixed length synthetic samples. The second phase of our experiments involves applying the pipeline consisting of the $Data_2$ implementation and the entire setup explained in the "Data" section. All experiments in this part were run with the Wasserstein GAN with gradient penalty setup. Table 3.3 synthesizes the achieved results.

Training procedure First, under the GAN formulation, Wasserstein training (with gradient penalty) outperforms its vanilla counterparts. Choosing between mixed or complete models does not have a major influence on the trend prediction accuracy, both techniques offering similar results. Lastly, batch normalization layers do not not hurt the model training anymore as reported in [4, 27].

Metrics High values of JSD mean that the model does not perform well. Low values, however, do not necessarily indicate good models, since this metric can be minimized when the model collapses to a single sample, marked with "*" int Table 3.3. Moreover, samples that follow the exact same path in the cumulative sum graph indicate that the generator model collapsed on a single set of values. We noticed that models that generate samples whose probability density function (PDF) matches the real samples' probability density function tend to offer good results. We also noticed that models that managed to fit the 4^{th} central moment, i.e., kurtosis, behave well on many levels. Additionally, the proposed dataset preparation technique helped in capturing autocorrelation by feeding cross-correlated samples at each iteration. Out of the proposed models, the clear winner is $VAE\_FC\ complete$, which outperforms every other model in almost every aspect.

Table 3.3 Qualitative and quantitative results (✓means that the property is met, whereas ✗the contrary).

Model	Regime	Mean	Var	Skew	Kurtosis	Autocorr	Heavy tail	Cluster Volatility	Cum Sum	Trend Ratio	JS	KL	K-S	EM
MLP_1	complete	✓	✓	✓	✗	✓	✗	✗	✓	✓	0.4511	237.8	0.2759	0.1254
MLP_1	up	✗	✗	✓	✗	✗	✗	✗	✓	✓	0.575	698.4	0.3546	0.1053
MLP_1	down	✗	✗	✓	✗	✓	✗	✗	✓	✓	0.4782	394	0.302	0.1103
MLP_1 [27, 4]	N/A	✗	✗	✓	✗	✗	✗	✗	N/A	N/A	0.5757	721.5	0.5757	0.114
MLP_2	complete*	✗	✗	✗	✗	✗	✗	✗	✗	✗	0.4845	276.9	0.3204	0.1196
MLP_2	up*	✓	✗	✗	✗	✗	✗	✗	✗	✗	0.6263	961.6	0.4019	0.1424
MLP_2	down*	✓	✗	✗	✗	✗	✗	✗	✗	✗	0.6841	677.4	0.4184	0.4029
MLP_2 [27, 4]	N/A	✓	✓	✓	✗	✓	✓	✗	N/A	N/A	0.0897	29.81	0.0301	0.0028
MLP_3	complete*	✓	✗	✗	✗	✗	✗	✗	✗	✗	0.5836	621.4	0.3774	0.2012
MLP_3	up*	✓	✗	✗	✗	✗	✗	✗	✗	✗	0.7216	1359	0.445	0.3872
MLP_3	down*	✗	✗	✗	✗	✗	✗	✗	✗	✗	0.6552	845.8	0.4115	0.413
MLP_3 [4]	N/A	✓	✓	✓	✗	✗	✓	✗	N/A	N/A	0.1235	128	0.0465	0.0088
$WMLP_3$ [4]	N/A	✓	✓	✓	✓	✓	✓	✗	N/A	N/A	0.1031	39.08	0.0323	0.003
MLP_4	complete	✓	✓	✓	✗	✓	✓	✗	✓	✓	0.1225	35.69	0.07936	0.00697
MLP_4	up	✓	✓	✗	✗	✓	✓	✗	✓	✓	0.2692	186.4	0.1641	0.01232
MLP_4	down	✓	✗	✗	✗	✓	✗	✗	✓	✓	0.6354	536.4	0.3845	0.281
MLP_4 [4]	N/A	✓	✓	✓	✗	✗	✗	✗	N/A	N/A	0.2095	99.89	0.131	0.0106
$FCGAN_1$	complete	✓	✗	✗	✗	✓	✗	✓	✗	✓	0.2947	140.1	0.1818	0.06226
$FCGAN_1$	up	✓	✓	✓	✗	✗	✗	✗	✓	✓	0.4541	395.2	0.2745	0.07718
$FCGAN_1$	down	✓	✓	✗	✗	✓	✗	✗	✓	✓	0.4595	287.7	0.3001	0.2058
$FCGAN_1$ [27, 4]	N/A	✗	✗	✗	✗	✗	✗	✗	N/A	N/A	0.2315	197.5	0.1709	0.0115
$WFCGAN_1$ [4]	N/A	✓	✓	✓	✗	✓	✓	✗	N/A	N/A	0.057	18.23	0.0359	0..0018
$FCGAN_2$	complete	✗	✓	✓	✗	✗	✓	✗	✓	✓	0.1592	42.44	0.1385	0.01779
$FCGAN_2$	up	✗	✗	✗	✗	✗	✗	✗	✗	✗	0.3776	256.9	0.2438	0.06838
$FCGAN_2$	down	✗	✗	✓	✗	✓	✗	✗	✓	✓	0.5569	381.8	0.3451	0.2074
$FCGAN_2$ [27, 4]	N/A	✓	✓	✓	✓	✗	✓	✗	N/A	N/A	0.0454	13.26	0.0178	0.0011
$WFCGAN_2$ [27, 4]	N/A	✓	✓	✗	✗	✗	✓	✓	N/A	N/A	0.0825	26.06	0.0387	0.0034
$FCGAN_3$	complete*	✓	✗	✗	✗	✗	✓	✗	✗	✗	0.1198	83.27	0.05527	0.00156
$FCGAN_3$	up*	✓	✗	✗	✗	✗	✓	✗	✗	✗	0.1576	267.4	0.07755	0.00376
$FCGAN_3$	down*	✓	✗	✗	✗	✗	✓	✗	✗	✗	0.2508	92.44	0.1136	0.08977
$FCGAN_3$ [4]	N/A	✗	✗	✗	✗	✗	✗	✗	N/A	N/A	0.5341	673.8	0.4007	0.1026
$FCGAN_4$	complete	✓	✓	✗	✗	✗	✓	✗	✓	✓	0.1324	82.87	0.0525	0.002
$FCGAN_4$	up	✓	✗	✓	✓	✗	✓	✗	✓	✓	0.1739	92.61	0.1051	0.0056
$FCGAN_4$	down	✓	✓	✗	✗	✗	✓	✗	✗	✗	0.1294	46.47	0.0902	0.0107
snFCGAN	complete*	✓	✓	✓	✗	✗	✓	✗	✗	✗	0.0666	5.613	0.0383	0.0039
snFCGAN	up*	✓	✓	✗	✗	✗	✗	✗	✗	✗	0.4882	719.4	0.3216	0.0525
snFCGAN	down*	✗	✗	✗	✗	✗	✗	✗	✗	✗	0.2985	140.1	0.2476	0..0475
snFCGAN [4]	N/A	✗	✗	✗	✗	✗	✓	✗	N/A	N/A	0.0953	23.53	0.0408	0.0031
FCGANmc [4]	N/A	✓	✓	✓	✓	✗	✓	✗	N/A	N/A	0.1101	78.16	0.0797	0.0032
GMMN_AE_FC	complete*	✗	✗	✗	✗	✗	✗	✗	✗	✗	0.1211	155.9	0.03031	0.00153
GMMN_AE_FC	up*	✗	✗	✗	✗	✗	✗	✗	✗	✗	0.1926	436.9	0.1069	0.00358
GMMN_AE_FC	down*	✗	✗	✗	✗	✗	✗	✗	✗	✗	0.2783	985.3	0.141	0.00951
GMMN_AE_MLP	complete*	✗	✗	✗	✗	✗	✗	✗	✗	✗	0.3671	1349	0.1937	0.0165
GMMN_AE_MLP	up*	✗	✗	✗	✗	✗	✗	✗	✗	✗	0.2576	1417	0.1237	0.0052
GMMN_AE_MLP	down*	✗	✗	✗	✗	✗	✗	✗	✗	✗	0.2278	822.4	0.0941	0.0072
VAE_FC	complete	✓	✓	✓	✓	✓	✓	✗	✓	✓	0.0929	14.26	0.0553	0.00566
VAE_FC	up	✓	✓	✓	✓	✓	✓	✗	✓	✓	0.07917	42.5	0.0273	0.00156
VAE_FC	down	✓	✓	✓	✓	✓	✓	✗	✓	✓	0.09543	30.97	0.0343	0.00357
VAE_MLP	complete	✓	✓	✓	✓	✗	✓	✗	✓	✓	0.1058	54.4	0.07054	0.00312
VAE_MLP	up	✓	✗	✓	✓	✓	✓	✗	✓	✓	0.06591	21.67	0.03615	0.00182
VAE_MLP	down	✓	✗	✓	✗	✗	✓	✗	✓	✓	0.09136	19.79	0.04364	0.00331

Prediction framework We tested 3 different networks in the prediction stage, namely a 2-layer bidirectional LSTM, ResNet-50, and a 4-layer MLP. We augmented the training dataset with synthetic data obtained with each of the models presented in Table 3.4 and tested the prediction algorithm on real data. The highest accuracies were obtained with the LSTM model so we only report them. We report the mean and max accuracies obtained over the 10 train-test split pairs. The maximum accuracies are on average 0.11% higher than the mean accuracies showing an important performance variation, confirming the volatility nature of stock markets. The baseline value was obtained by training and testing the prediction algorithm on real data only.

Table 3.4 Trend prediction accuracies on augmented dataset.

Model	Mean accuracy	Max accuracy
MLP_1 *complete*	50.12%	50.26%
MLP_1 *mixed*	50.30%	50.58%
MLP_2 *complete*	50.41%	50.56%
MLP_2 *mixed*	50.40%	50.48%
MLP_3 *complete*	50.38%	50.43%
MLP_3 *mixed*	50.40%	50.50%
MLP_4 *complete*	50.34%	50.45%
MLP_4 *mixed*	50.27%	50.38%
$FCGAN_1$ *complete*	50.27%	50.40%
$FCGAN_1$ *mixed*	50.35%	50.45%
$FCGAN_2$ *complete*	50.40%	50.54%
$FCGAN_2$ *mixed*	50.11%	50.27%
$FCGAN_3$ *complete*	50.37%	50.43%
$FCGAN_3$ *mixed*	50.28%	50.37%
$FCGAN_4$ *complete*	50.39%	50.46%
$FCGAN_4$ *mixed*	50.32%	50.41%
sn_FCGAN complete	50.31%	50.48%
sn_FCGAN mixed	50.39%	50.44%
GMMN_AE_FC complete	50.40%	50.44%
GMMN_AE_FC mixed	50.42%	50.52%
GMMN_AE_MLP complete	50.40%	50.48%
GMMN_AE_MLP mixed	50.41%	50.57%
VAE_FC complete	50.31%	50.41%
VAE_FC mixed	50.29%	50.42%
VAE_MLP complete	50.18%	50.28%
VAE_MLP mixed	50.22%	50.32%
Baseline	50.04%	

3.3.4 Conclusions

We proposed a complex framework for generating realistic financial time series. We proposed a new way of extracting batches of data from the training set, adapted to the particularity of financial time series. We investigated 3 major classes of generative models with various model composition, setups, hyperparameters, training frameworks, and data regimes. We examined different qualitative and quantitative metrics and tested the dataset augmentation ability on real data, under a complex prediction scenario.

We identified the variational autoencoder framework as being the best suited model for the current task. Finally, we stress the need of a metric or validation framework that can harness both objective and subjective properties under a single quantifiable value.

Chapter 4

Summary of Contributions and Future Work

4.1 Summary of contributions

In this section we will present the obtained results, on a per-chapter basis. Since these results have been discussed at large in their corresponding sections, we will only briefly introduce them here.

In Chapter 2 we discussed several approaches for image retrieval on scarce datasets. This work has been divided in two parts as follows.

- Section 2.2 covers our progress over 3 years (2017, 2018, and 2019) at a lifelog benchmarking competition.
  - In 2017 [5] we proposed a novel approach that automatically analyzes the similarity between a text query and a set of images for which concepts are available.
  - In 2018 [6] we proposed a more sophisticated approach that involved parsing the text query for critical information regarding activity, location, time, and date. Corroborating these with several detectors results in a shortlist of relevant images.
  - In 2019 [7] we opted for a more aggressive filtration of the dataset, excluding images that did not meet a set of restrictions related to blurriness, activity, location, time, time zone, person count, etc., and applied a weighted scoring mechanism on the similarities between relevant concepts extracted from the search query and those extracted from the images.
- Section 2.3 presents the system that we proposed [9] to solve the unattended luggage detection. We run an object detector on every image from the dataset and detect luggage that are isolated from persons. Then, we compare the isolated

object's feature vector to other similar objects in the CCTV stream in order to determine who left it there. After finding a suspect, we also search the CCTV feed to find where this person has been seen before.

In Chapter 3 we approach the scarce data problem from a different perspective. Instead of finding ways of adapting our algorithms to limited data availability we propose different methods of augmenting the dataset by using generative models.

- Section 3.2 presents the method we propose [8] for controlling the logos that a DCGAN can hallucinate. We train a DCGAN framework to generate logos, apply a backpropagation mechanism to retrieve an approximation of the latent code that generated the logos, and augment the original dataset with reconstructed versions of the original logos.

- Section 3.3 addresses our approach [4] on the problem of financial time-series generation. We propose several generative architectures (GAN, VAE, GMMN), with an entire set of variations for each type, to generate realistic time-series in the context of financial markets. We propose new ways of generating and mixing synthetic samples such that they appear to be realistic. We also investigate how validation can be performed on this type of data.

4.2 Original contributions

In this section I present the original contributions that I had on each individual publication.

In [Ch1], [Ch2], [Ch3], [Ch4], [C7], and [C9] I was part of the organizing committee of the ImageCLEF benchamrking campaigns, where I was involved in the competitions' organization, in charge of the participants' registration.

In [J1] I designed and implemented the proposed generative framework for financial time-series. This consisted of extending our previous research [4] to new generative models (e.g. AE_GMMN, VAEs, WGANs etc.), adapting them to the financial setup, implementing new metrics, implementing the regime split and mixing of models, and running the entire pipeline. Afterwards, I also took part in the prediction stage, where I ran a considerable part of the presented experiments and centralized the final results.

In [J2] I proposed to adapt the backpropagation approach to DCGAN in order to control sample generation. I also implemented the entire system and carried out the required validation. This work was conducted during an ERASMUS internship at CEA-List, Palaiseau, France.

In [C8] I designed and implemented the proposed generative framework for financial time-series. This consisted of proposing several network architectures, adapting them to the financial setup, finding possible metrics for validating them, and run the entire pipeline.

In [C6] I proposed and implemented the idea of using the object detector's features for solving all 3 involved subtasks (abandoned luggage detection, suspect detection, and suspect re-identification). I was also responsible with the validation dataset gathering process which consisted of designing and implementing a task-specific GUI for person annotations, coordinating the annotators, centralizing all data, and managing the entire process.

In [C5] I was involved in the person detection feature extraction process.

In [C4] and [C2] I proposed and implemented several filtering procedures that would eliminate most of the uninformative images from a lifelog dataset. Then, I ran several concept detectors and designed a rule that would rank the images based on the scores that were obtained by the concept detectors. The highest-scoring images would correspond to the search query.

In [C3] I was involved in the design of the autoencoder that was used for feature extraction of electroencephalogram signals.

In [C1] I proposed and implemented a method that establishes a connection between the textual description of several search queries and the output of a concept detector ran on a dataset of lifelog images. This connection relies on NLP methods that examine the similarity between different words which are part of a large dictionary. Empirical thresholds were set in order to limit the vast amount of image proposals for each search query.

4.3 Future Perspectives

This thesis focused on the way several applications can deal with scarce datasets. Naturally, we will push forward the efforts that we carried out in the object detection field and working with scarce data by concentrating on few-shot learning and few-shot object detection, in particular. We will also benefit from the experience that we acquired with generative models by augmenting weakly represented classes. This will also help with the class imbalance problem that is so often encountered in deep learning applications, especially in real-world scenarios.

4.4 Publications

Book Chapters

[Ch4] Ionescu, B., Müller, H., Péteri, R., ..., **Dogariu, M.**, ..., Deshayes, J. (2021). Overview of the ImageCLEF 2021: Multimedia Retrieval in Medical, Nature, Internet and Social Media Applications. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, Cham. Springer International Publishing. [17]

[Ch3] Ionescu, B., Müller, H., Péteri, R., ..., **Dogariu, M.**, Ştefan, L. D., Constantin, M. G., Deshayes, J., and Popescu, A. (2021). The 2021 ImageCLEF benchmark: Multimedia retrieval in medical, nature, internet and social media applications. In *Advances in Information Retrieval*, pages 616–623, Cham. Springer International Publishing. [16]

[Ch2] Ionescu, B., Müller, H., Péteri, R., ..., **Dogariu, M.**, Ştefan, L. D., and Constantin, M. (2020). Overview of the ImageCLEF 2020: Multimedia retrieval in medical, lifelogging, nature, and internet applications. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 311–341, Cham. Springer International Publishing. [15]

[Ch1] Ionescu, B., Müller, H., Péteri, R., ..., **Dogariu, M.**, Ştefan, L. D., and Constantin, M. G. (2020). ImageCLEF 2020: Multimedia retrieval in lifelogging, medical, nature, and internet applications. In *Advances in Information Retrieval*, pages 533–541, Cham. Springer International Publishing. [18]

Journals

[J1] **Dogariu, M.**, Ștefan, L.-D., Boteanu, B. A., Lamba, C., Kim, B. and Ionescu, B. (2021). Generation of Realistic Synthetic Financial Time-Series. In *ACM Transactions onMultimidia Computing Communications and Applications*. Paper under review. [10]

[J2] **Dogariu, M.**, Le Borgne, H., and Ionescu, B. (2021b). Backpropagation aided logo generation using generative adversarial networks. In *University Politehnica of Bucharest Scientific Bulletin Series C-Electrical Engineering And Computer Science*, 83(2):59–70. [8]

Conferences

[C9] Berari, R., Tauteanu, A., Fichou, D., Brie, P., **Dogariu, M.**, Ştefan, L.-D., Constantin, M. G. and Ionescu, B. (2021). Overview of ImageCLEFdrawnUI 2021: The Detection and Recognition of Hand Drawn and Digital Website UIs Task. In *Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum.*, Bucharest, Romania, September 21-24, volume 2936. [1]

[C8] **Dogariu, M.**, Ștefan, L.-D., Boteanu, B. A., Lamba, C., and Ionescu, B. (2021).Towards realistic financial time series generation via generative adversarial learning. In *2021 29th European Signal Processing Conference (EUSIPCO)*. [4]

[C7] Fichou, D., Berari, R., Brie, P., **Dogariu, M.**, Ştefan, L.-D., Constantin, M. G. and Ionescu, B. (2021). In *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum.*, Thessaloniki, Greece, September 22-25, volume 2696. [11]

[C6] **Dogariu, M.**, Ștefan, L.-D., Constantin, M. G., and Ionescu, B. (2020). Human-object interaction: Application to abandoned luggage detection in video surveillance scenarios. In *2020 13th International Conference on Communications (COMM)*, pages 157–160. IEEE. WOS:000612723900028. [9]

[C5] Ștefan, L.-D., Abdulamit, Ș., **Dogariu, M.**, Constantin, M. G., and Ionescu, B. (2020). Deep learning-based person search with visual attention embedding. In *2020 13th International Conference on Communications (COMM)*, pages 303-308. IEEE. WOS:000612723900053. [3]

[C4] **Dogariu, M.** and Ionescu, B. (2019). Multimedia lab @ ImageCLEF 2019 lifelog moment retrieval task. In *Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum*, Lugano, Switzerland, September 9-12, volume 2380. CEUR-WS.org. [7]

[C3] Tăuţan, A.-M., **Dogariu, M.**, and Ionescu, B. (2019). Detection of Epileptic Seizures using Unsupervised Learning Techniques for Feature Extraction. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 2377-2381). IEEE. WOS:000557295302184. [28]

[C2] **Dogariu, M.** and Ionescu, B. (2018). Multimedia lab @ ImageCLEF 2018 lifelog moment retrieval task. In *Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum*, Avignon, France, September 10-14, volume 2125. CEUR-WS.org. [6]

[C1] **Dogariu, M.** and Ionescu, B. (2017). A textual filtering of HOG-based hierarchical clustering of lifelog data. In *Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum*, Dublin, Ireland, September 11-14, volume 1866. CEUR-WS.org. [5]

International Research Projects

[IntR2] *October 2020 - present*: **PhD Student** at University Politehnica of Bucharest, in the H2020 AI4Media "A European Excellence Centre for Media, Society and Democracy" project, owner CERTH, Greece, partner University "Politehnica" of Bucharest, axis H2020 ICT-48-2020 / Towards a vibrant European network of AI excellence centres.

[IntR1] *February 2017 - April 2019*: **Trainer** at University Politehnica of Bucharest, in the UMETECH project "University and Media Technology for Cultural Heritage", funded by the European Commission, ID 574105-EPP-1-2016-1-IT-EPPKA2-CBHE-JP.

National Research Projects

[NR4] *June 2020 - present*: **PhD Student** at University Politehnica of Bucharest, in the SMARTRetail "Enhancing and Improving Customer Experience and Services in Supermarkets via SMART Artificial Intelligence Powered Systems" project, owner Softrust Vision Analytics, partner University "Politehnica" of Bucharest, ID PN-III-P2-2.1-PTE-2019-0055.

[NR3] *June 2020 - present*: **PhD Student** at University Politehnica of Bucharest, in the GRAVI "Virtual Guardian: Artificial Intelligence Powered Multi-Sensor System for Automatic Securing of Areas of Interest", owner Softrust Vision Analytics, partner University "Politehnica" of Bucharest, ID PN-III-P2-2.1-PTE-2019-0570.

[NR2] *May 2017 - April 2020*: **PhD Student** at University Politehnica of Bucharest, in the SPIA-VA project ("Intelligent Systems for Video and Audio Analysis - Technologies and Innovative Video Systems for Person Re-identification and Analysis of Dissimulated Behavior", project coordinator University Politehnica of Bucharest, ID PN-III-P2-2.1-SOL-2016-02-000.

[NR1] *January 2017 - June 2018*: **PhD Student** at University Politehnica of Bucharest, in the SPOTTER project ("Intelligent Real-time Surveilance System with Specific Regions Detection Integrated on IP Cameras"), funded by the Romanian Government through UEFISCDI, project coordinator University Politehnica of Bucharest, ID PN-III-P2-P2.1-PED-2016-1065.

Industrial Research Projects

[IndR3] *March 2020 - October 2020*: **PhD Student** at University Politehnica of Bucharest, in the Keysight 1 "Machine Learning Techniques for Generating Network Traffic Data" project, owner University "Politehnica" of Bucharest, Research Center CAMPUS, beneficiary Keysight Technologies Romania, ID Keysight 1/23-03-2020.

[IndR2] *April 2020 - July 2020*: **PhD Student** at University Politehnica of Bucharest, in the Hana 2 "Financial Data Augmentation and Forecasting Using Advanced AI Techniques" project, owner "Politehnica" Research, Development and Innovation Institute, beneficiary Hana Institute of Technology, Republic of Korea, ID Hana 2/01-04-2020.

[IndR1] *July 2019 - December 2019*: **Research Assistant** at University Politehnica of Bucharest, in the Hana 1 "Machine Learning Techniques for the Processing and Analysis of Financial Data" project, owner "Politehnica" Research, Development and Innovation Institute, beneficiary Hana Institute of Technology, Republic of Korea, ID Hana 1/22-07-2019.

References

[1] Berari, R., Tauteanu, A., Fichou, D., Brie, P., Dogariu, M., Ştefan, L. D., Constantin, M. G., and Ionescu, B. (2021). Overview of ImageCLEFdrawnUI 2021: The detection and recognition of hand drawn and digital website uis task. In *Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum.*, volume 2936 of *CEUR Workshop Proceedings*, pages 1121—1132, Bucharest, Romania. CEUR-WS.org <http://ceur-ws.org>.

[2] Builders, A. I. D. N. N. I. A. (2014). Unattended baggage detection using deep neural networks in intel® architecture. Technical report, Intel Corporation.

[3] Ştefan, L.-D., Abdulamit, c., Dogariu, M., Constantin, M. G., and Ionescu, B. (2020). Deep learning-based person search with visual attention embedding. In *2020 13th International Conference on Communications (COMM)*, pages 303–308.

[4] Dogariu, M., Ştefan, L.-D., Boteanu, B. A., Lamba, C., and Ionescu, B. (2021a). Towards realistic financial time series generation via generative adversarial learning. In *2021 29th European Signal Processing Conference (EUSIPCO)*.

[5] Dogariu, M. and Ionescu, B. (2017). A textual filtering of hog-based hierarchical clustering of lifelog data. In Cappellato, L., Ferro, N., Goeuriot, L., and Mandl, T., editors, *Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, September 11-14, 2017.*, volume 1866. CEUR-WS.org.

[6] Dogariu, M. and Ionescu, B. (2018). Multimedia lab @ imageclef 2018 lifelog moment retrieval task. In Cappellato, L., Ferro, N., Nie, J.-Y., and Soulier, L., editors, *Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018.*, volume 2125. CEUR-WS.org.

[7] Dogariu, M. and Ionescu, B. (2019). Multimedia lab @ imageclef 2019 lifelog moment retrieval task. In Cappellato, L., Ferro, N., Losada, D. E., and Müller, H., editors, *Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019.*, volume 2380. CEUR-WS.org.

[8] Dogariu, M., Le Borgne, H., and Ionescu, B. (2021b). Backpropagation aided logo generation using generative adversarial networks. *University Politehnica of Bucharest Scientific Bulletin Series C-Electrical Engineering And Computer Science*, 83(2):59–70.

[9] Dogariu, M., Stefan, L.-D., Constantin, M. G., and Ionescu, B. (2020). Human-object interaction: Application to abandoned luggage detection in video surveillance scenarios. In *2020 13th International Conference on Communications (COMM)*, pages 157–160. IEEE.

[10] Dogariu, M., Ștefan, L.-D., Boteanu, B., Lamba, C., Kim, B., and Ionescu, B. (2021c). Generation of realistic synthetic financial time-series. *ACM Transactions on Multimedia Computing Communications and Applications*. Paper under review.

[11] Fichou, D., Berari, R., Brie, P., Dogariu, M., Ştefan, L. D., Constantin, M. G., and Ionescu, B. (2020). Overview of ImageCLEFdrawnUI 2020: The detection and recognition of hand drawn website uis task. In *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum.*, volume 2696 of *CEUR Workshop Proceedings*, Thessaloniki, Greece. CEUR-WS.org <http://ceur-ws.org>.

[12] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[13] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

[14] He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

[15] Ionescu, B., Müller, H., Péteri, R., Abacha, A. B., Datla, V., Hasan, S. A., Demner-Fushman, D., Kozlovski, S., Liauchuk, V., Cid, Y. D., Kovalev, V., Pelka, O., Friedrich, C. M., García Seco de Herrera, A., Ninh, V.-T., Le, T.-K., Zhou, L., Piras, L., Riegler, M., Halvorsen, P., Tran, M.-T., Lux, M., Gurrin, C., Dang-Nguyen, D.-T., Chamberlain, J., Clark, A., Campello, A., Fichou, D., Berari, R., Brie, P., Dogariu, M., Ştefan, L. D., and Constantin, M. G. (2020a). Overview of the imageclef 2020: Multimedia retrieval in medical, lifelogging, nature, and internet applications. In Arampatzis, A., Kanoulas, E., Tsikrika, T., Vrochidis, S., Joho, H., Lioma, C., Eickhoff, C., Névéol, A., Cappellato, L., and Ferro, N., editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 311–341, Cham. Springer International Publishing.

[16] Ionescu, B., Müller, H., Péteri, R., Abacha, A. B., Demner-Fushman, D., Hasan, S. A., Sarrouti, M., Pelka, O., Friedrich, C. M., de Herrera, A. G. S., Jacutprakart, J., Kovalev, V., Kozlovski, S., Liauchuk, V., Cid, Y. D., Chamberlain, J., Clark, A., Campello, A., Moustahfid, H., Oliver, T., Schulz, A., Brie, P., Berari, R., Fichou, D., Tauteanu, A., Dogariu, M., Stefan, L. D., Constantin, M. G., Deshayes, J., and Popescu, A. (2021a). The 2021 imageclef benchmark: Multimedia retrieval in medical, nature, internet and social media applications. In Hiemstra, D., Moens, M.-F., Mothe, J., Perego, R., Potthast, M., and Sebastiani, F., editors, *Advances in Information Retrieval*, pages 616–623, Cham. Springer International Publishing.

[17] Ionescu, B., Müller, H., Péteri, R., Ben Abacha, A., Sarrouti, M., Demner-Fushman, D., Hasan, S. A., Kozlovski, S., Liauchuk, V., Dicente, Y., Kovalev, V., Pelka, O., de Herrera, A. G. S., Jacutprakart, J., Friedrich, C. M., Berari, R., Tauteanu, A., Fichou, D., Brie, P., Dogariu, M., Ştefan, L. D., Constantin, M. G., Chamberlain, J., Campello, A., Clark, A., Oliver, T. A., Moustahfid, H., Popescu, A., and Deshayes-Chossart, J. (2021b). Overview of the ImageCLEF 2021: Multimedia retrieval in medical, nature, internet and social media applications. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, Proceedings of the 12th International Conference of the CLEF Association (CLEF 2021), Bucharest, Romania. LNCS Lecture Notes in Computer Science, Springer.

[18] Ionescu, B., Müller, H., Péteri, R., Dang-Nguyen, D.-T., Zhou, L., Piras, L., Riegler, M., Halvorsen, P., Tran, M.-T., Lux, M., Gurrin, C., Chamberlain, J., Clark, A., Campello, A., Seco de Herrera, A. G., Ben Abacha, A., Datla, V., Hasan, S. A., Liu, J., Demner-Fushman, D., Pelka, O., Friedrich, C. M., Dicente Cid, Y., Kozlovski, S., Liauchuk, V., Kovalev, V., Berari, R., Brie, P., Fichou, D., Dogariu, M., Stefan,

L. D., and Constantin, M. G. (2020b). Imageclef 2020: Multimedia retrieval in lifelogging, medical, nature, and internet applications. In Jose, J. M., Yilmaz, E., Magalhães, J., Castells, P., Ferro, N., Silva, M. J., and Martins, F., editors, *Advances in Information Retrieval*, pages 533–541, Cham. Springer International Publishing.

[19] Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *International Conference on Learning Representation (ICLR)*.

[20] Lecun, Y. (1987). *Modeles connexionnistes de l'apprentissage*. PhD thesis, These de Doctorat, Universite Paris.

[21] Li, Y., Swersky, K., and Zemel, R. (2015). Generative moment matching networks. In *International Conference on Machine Learning*, pages 1718–1727. PMLR.

[22] Lipton, Z. C. and Tripathi, S. (2017). Precise recovery of latent vectors from generative adversarial networks. In *International Conference on Learning Representation (ICLR) Workshop*.

[23] Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. In Bengio, Y. and LeCun, Y., editors, *4th International Conference on Learning Representations, ICLR*.

[24] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

[25] Romberg, S., Pueyo, L. G., Lienhart, R., and Van Zwol, R. (2011). Scalable logo recognition in real-world images. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, pages 1–8.

[26] Sage, A., Agustsson, E., Timofte, R., and Van Gool, L. (2018). Logo synthesis and manipulation with clustered generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5879–5888.

[27] Takahashi, S., Chen, Y., and Tanaka-Ishii, K. (2019). Modeling financial time-series with generative adversarial networks. *Physica A: Statistical Mechanics and its Applications*, 527:121261.

[28] Tăuţan, A.-M., Dogariu, M., and Ionescu, B. (2019). Detection of epileptic seizures using unsupervised learning techniques for feature extraction. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2377–2381.

[29] Xiao, T., Li, S., Wang, B., Lin, L., and Wang, X. (2017). Joint detection and identification feature learning for person search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3415–3424.