A domain adaptation of person re-identification with similar apparel

Thesis - BallJames

Graduate
Fabian Mijsters
432601@student.saxion.nl

Company supervisor
Deepak Viswanathan
d.viswanathan@scisports.com

Saxion supervisor
Evert Duipmans
e.f.duipmans@saxion.nl

Version: 0.7
Date: 14-06-2020

Version control

Version	Status	Remarks	Date
0.1	Draft	Initial version	21-03-2020
0.2	Draft	Feedback Saxion Supervisor	20-04-2020
0.3	Draft	Feedback Saxion Supervisor	06-05-2020
0.4	Draft	Feedback Saxion Supervisor and BallJames Supervisor	13-05-2020
0.5	Draft	Feedback Saxion Supervisor	27-05-2020
0.6	Concept version	Concept deadline	02-06-2020
0.7	Final version	Final deadline	14-06-2020

Table 1 - Version control

Abstract

This thesis explores person re-identification in the football domain. BallJames generates tracking data by tracking football players during a match. When an occlusion happens between multiple players, BallJames loses track of the player. Person re-identification could solve this problem by comparing the appearances of the players. A standard person re-identification model is not able to differentiate between players of the same team due to the similar apparel. By training a new person re-identification model on domain specific training data this problem is solved. The domain specific training data is collected using a newly created data extractor which can create semi automatic labeled training data in a time feasible manner.

Table of contents

1. Introduction

Winning is more important than ever in football. The cash prices are increasing every year and so is the worth of the players. Football clubs will do anything to get an edge over their opponents. Data analysis is used to improve the strategy of the team and the form of individual players. The recordings of matches are used to better analyse a team. Analysing these recordings can take a long time. Because of this reason optical tracking data is very valuable for a team's video analyst. The media and betting sector profit from tracking data by using the data to write articles and by predicting the most profitable odds respectively. This data often consists of x,y,z coordinates of a player in a frame and in a 3d world representation.

BallJames created an autonomous optical tracking solution. BallJames uses the recordings of football matches to generate tracking data. This tracking data can be used in a number of fields including performance analysis, betting and the media. The value of tracking data is dependent on the accuracy and the completeness(is every player tracked in each frame). Tracking in BallJames's domain is the act of following a player over the course of a match.

Tracking can be done with multiple techniques including wearable devices like gps trackers, optical tracking, and by manual annotation. BallJames uses optical tracking which is a category of software based tracking. A video consists of a number of frames in a sequence. Optical tracking relies on detecting players in each frame. A detection represents simply the x and y coordinates of the object and its width and height. By comparing and combining a detection in a frame to the detection in the next frame a player can be tracked. Seeing that there are 22 players on pitch during a game a lot of players need to be tracked independently and at the same time. This is a hard problem since you need to know which player is which and you can't combine the detections randomly. This problem can be solved in multiple ways. BallJames checks how much detections in sequential frames overlap. If they overlap for a set amount the detections are put together in the same "track".

Tracking is a common act and it's used in a lot of areas including some shopping malls, airports or high end shops. However the tracking problem BallJames faces is unique since there are a lot of similar appearances on the pitch.. A standard football match includes 22 players of which two groups of ten wear the same clothes, the other two players are the keepers and wear their own separate kit.

The tracking accuracy degrades whenever occlusions occur. Occlusions make it hard for the tracking application to detect the player. After an occlusion the track of a player is broken since there is no overlap between the new detection and the last detection of a player. Person re-identification might be able to link the track before an occlusion took place to the track after an occlusion.

Person re-identification is a promising new route to tracking. Person re-identifications popularity is due to the fact that deep learning models are improving day-by-day. A re-identification model is a kind of deep learning model which can encode a person's appearance in a way that a computer can understand. This description can be compared to the "appearance descriptor" of another person/player. This comparison leads to a simple number which represents the similarity between two images of players. The description is based on multiple appearance features including but not limited to the clothes, the pose and illumination. Seeing that appearance in the domain of BallJames is constricted, out-of-the box person re-identification models might not perform well enough. This means that a person re-identification model needs to be trained on domain specific data.

Deep learning models such as a person re-identification model are trained using a lot of data. The model predicts features based on a large set of simple operations. These operations are automatically modified when the predicted feature/description is not useful. This process is repeated for all the images in the dataset until the results are good enough to be used.

The result of the training of a deep learning model depends heavily on the quality of the data. The dataset should represent what a deep learning model should learn. A larger dataset generally has better results than small datasets. This is mostly because the model sees different kinds of data and that helps the model generalize. Creating such a dataset by hand is not time and cost efficient.

To train a person re-identification model a dataset is needed which contains a set of identities. One identity is a single person, it should only contain images of that person in multiple poses. If a big enough dataset is created, it should lead to a robust model which generalises well on people it did not see before.

The person re-identification model that will be implemented at BallJames needs to be trained on a dataset that represents the domain of BallJames. This dataset will contain multiple identities(players) with each identity containing only images of that one person in multiple poses. The quality of the tracking of BallJames is high enough which means that this tracking can be used to semi-automatically create a dataset for this specific problem or following problems which require a dataset. A new program needs to be written which can extract the dataset from the tracking data and save it in such a way that it can be used for training. This focuses mainly on the person re-identification model and the scalable agile data generator.

1.1 Research questions

Can re-identification be performed on people who are similar in appearance and how can it be implemented?

1. What are the most applicable currently available solutions to person re-identification?
2. How is training data structured in existing re-identification datasets and how should this be implemented for our problem?
3. Can we rely on existing multi view tracking data to generate partially labeled tracking data in a time feasible way?
4. Since we're in an apparel constricted scenario we can't rely on a standard global appearance descriptor, will domain specific training data solve this?
5. What distance metrics should be used when testing the deep learning models?
6. What evaluation methods are important when testing a deep learning model for this specific use case?

2. Background

2.1 Person Re-Identification

Person re-identification is another way of tracking a person. In most cases person re-identification is used as an extension to another version of logic based tracking. Person re-identification can be implemented in more than one way. A common way of implementing person re-identification is by creating an appearance descriptor of a certain player inside of a boundingbox. An appearance descriptor is a vector of *n* possible features which describes the appearance of the player. The appearance descriptor is generated using a trained neural network. This appearance descriptor can be compared to other appearance descriptors using certain distance functions including but not limited to cosine similarity and mahalanobis distance. The result of the distance function can be interpreted as a possibility of belonging to the same soft identity. In the ideal scenario the appearance descriptor created by a re-identification model when compared using a distance function produces a high distance between appearance descriptors of different persons and an as small as possible distance between appearance descriptors of the same person.

3. BallJames

BallJames offers two products. A pro package which is meant for professional football clubs with an extensive need and budget for analysing their team. The pro package consists of fourteen properly calibrated 4k cameras. The cameras are installed in a way which ensure optimal coverage of the pitch. The second product is the SMART package. The SMART product uses the cameras already installed around the pitch. Since the cameras are not installed by BallJames ideal pitch coverage can't be guaranteed.

3.1 Product flow

The product flow of the two products is quite similar. For sake of compactness the two products will be explained as one. All the parts explained below are stored and accessible via Kafka topics. Kafka is an open-source stream-processing software platform. Only the parts which are inside of the scope of this assignment will be explained.

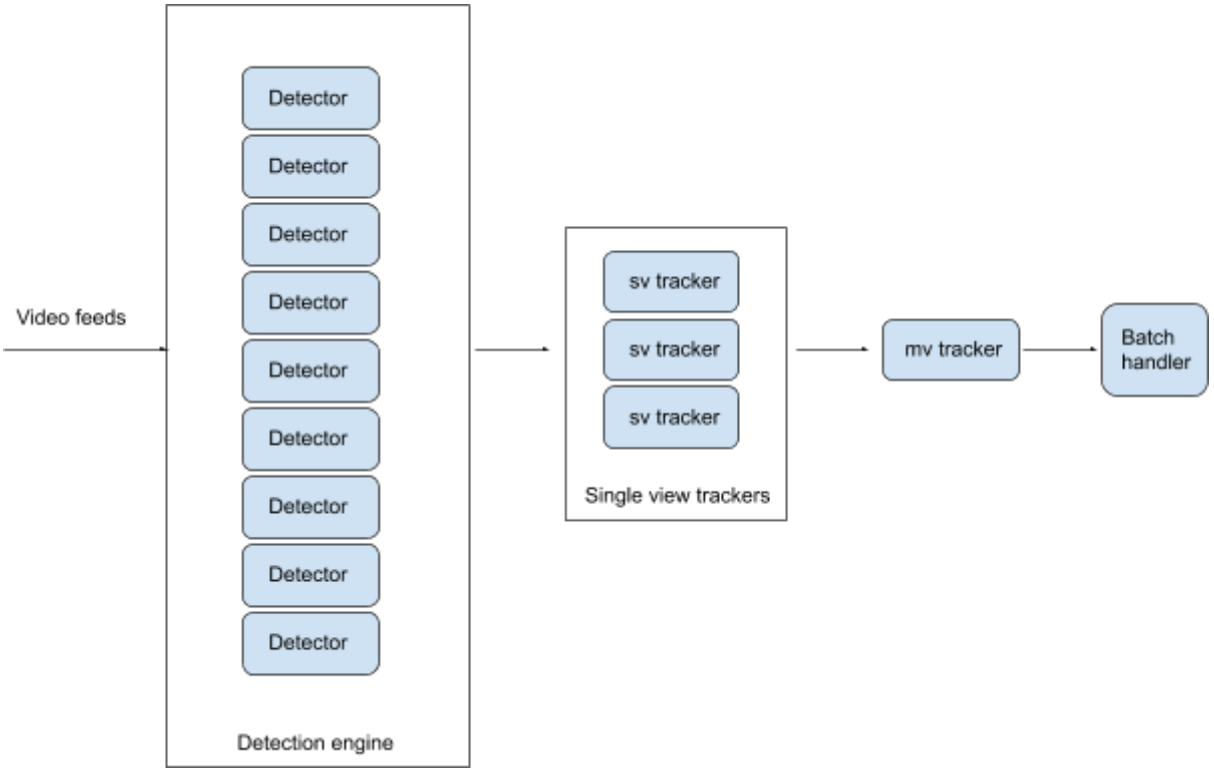

Figure 1 - A simplified version of the BallJames pipeline

3.1.1 Detection engine

The detection engine handles the initial detection of players. The detection engine is fed a video stream for each camera connected to the system. Detections are generated for each stream individually. These detections simply consist of a list of bounding boxes corresponding to each player. A bounding box is simply a rectangle around a player and is most commonly passed as x and y coordinates with a width and a height. This collection of

detections is sent to the single view tracker. The single view tracker tracks each player individually.

Figure 2 - Example of a single detection

3.1.2 Single view tracker

The single view tracker's job is to track each player. The single view tracker compares and combines bounding boxes that appear in sequential frames. This process results in a simple list for each "track" filled with all the bounding boxes belonging to a single player corresponding to a frame. The occlusion problem mentioned above takes place in the single view tracker. This is also the place where the solution will be implemented. A team classifier and number recognition model is also run on these tracks. The goal of these models is to detect to which team the player belongs and what his jersey number is. This information can be used to identify a player. The tracks in combinations with the identity information are sent to the multi view tracker. It's important to note that these tracks are generated for each camera feed which means that if there are fourteen cameras, there should be around 14 tracks for each player in the ideal scenario.

Figure 3 - Example of part of a single view track

3.1.3 Multi view tracker

The multi view tracker combines the multiple tracks of each player into a single "multi view track". A single multi view track consists of each track of the same player at the same

moment in time. These single view tracks are linked by using a technique which uses the real world coordinates of a bounding box in combination with the predicted identity of a player. This technique is outside the scope of this thesis but important to note. A correct identification of a player's team and number is needed to be able to use this technique. It is important to know which player is in a certain single view track. In case of the pro package, this will often happen seeing that there are 14 different angles available for every player. In the SMART package however, there can be as few as one camera. There might not be a clear view of the jersey number of a player if there is only one camera available. When there is no identity information available a track cannot be linked to a player and thus decreases the accuracy of the tracking data. Person re-identification should help to link an unidentified track to a player.

These newly created multi view tracks are sent to the batch handler. After the multi view tracker there are no longer multiple input streams but instead a single input containing all the information of all the cameras.

Figure 4 - Example of a frame from a single view track from one player in a multi view track

3.1.4 Batch handler

The batch handler takes every multi view track and averages the multiple tracks for one player to a single track which should have the most accurate position and predicted identity for each player. The batch handler tracks are the tracks that are used in the creation of the semi automatically labeled training data.

Figure 5 - Example of an identified player in the batch handler which is aggregated from all the cameras

4. Assignment

4.1 Training data generator

The assignment consists of two steps. The goal of the first step is creating a semi-automatic way of downloading and extracting training data from the available kafka topics containing the data from each step in the pipeline. The training data generator can be used in step two of the assignment and whenever BallJames wants to improve or experiment on this in the future. Since the training data needs to conform to the requirements of the model and the specific model is not yet determined, the training data generator needs to be built in such a way that it is customizable to the specific needs of a model. Because of the nature of the multi view tracker it is possible to generate partly labeled training data. The multi view tracker in most cases correctly identifies players and uses these identities to link single view tracks together. The linked single view tracks can in turn be used to generate training data from. Generating training data this way removes the need for extensive manual labeling and thus saves time and money. The data generator needs to be scalable to different camera setups. The data generator also needs to be easily configurable.

4.2 Re-identification model

The second step of the assignment is focused on creating a re-identifying model using the training data generator built in part one. In the second part of the assignment a research report will be created. The research report will be used as a base to either implement an existing re-identification model or create a new model based on existing techniques shaped to the specific needs of BallJames.

The second step of the assignment can in turn be split into two separate problems:

1. Short term re-identification
2. Long term re-identification

4.2.1 Short term re-identification

Short term re-identification is the re-identification of single views tracks that are broken for a short amount of time. To qualify for short term re-identification a broken track is judged based on the distance the player travelled since the last detection and the time that has passed since the last detection. The specifics will be decided during the testing phase of a model because this will be specific to the model and cannot be optimised without seeing the results of the values. For short term re-identification a soft identity will most likely be used. Since there is no need for true identity recognition. Recognising that two detected players are the same player is enough to link single view tracks together.

4.2.2 Long term re-identification

Long term re-identification entails re-identifying a player in any place and time in a match from a bank of appearance information of a player. A successful long term re-identification model is able to correctly re-identify players taken from a random frame of video from a saved bank of appearance information of a player. To be able to correctly re-identify a player over a relatively long time and distance, a hard identity is needed. A hard identity is knowing who the player is by team and number.

For this assignment the focus will be on the short term re-identification model. The long term re-identification model will be an extension of the short term re-identification and will only be worked on when the short term re-identification model is tested and accepted by BallJames

5. Projectorganization

The assignment is divided into four phases. These phases will help organize the assignment in logical steps and bring structure to the project. The four phases are: Preparation phase, research phase, development phase and the conclusion phase.

5.1 Day-to-day

5.1.1 Definition of done

A task will be done when all the requirements of a task are met and the task is reviewed and accepted by the company supervisor. A task can be removed when it is clear that the task is no longer needed. The removal of a task is also done in consultation with the company supervisor. When a task is able to be unit tested and thus is not a research task or a deep learning oriented task, A unit test will be written for the component and the task is done when the tests are completed positively.

5.1.1.1 Quality insurance

To ensure the written code is up to standards with the company code. The code will be compared to the code that is available in the different company repositories. At BallJames there are some libraries available that solve common problems when working with the tracking data, kafka and images. To ensure the quality of the code these libraries will be used.

5.1.2 Kanban

The kanban method will be applied in the development phase. The daily kanban part of the kanban process will be applied during each phase of the project and will be together with the BallJames development team.

5.1.2.1 Daily kanban

A daily kanban will be conducted every day the graduate is at BallJames. During the daily kanban the graduate will report what the graduate did the day before and what the graduate will be doing today. This will also function as short updates for the company supervisor. During the stand up the company supervisor can intervene if the supervisor notices any possible improvements or better courses of action.

5.1.2.2 To Do

Before the research phase the backlog will be filled with all the tasks that either will or will not be delivered. These tasks are added to the backlog because the details aren't clear yet.
The to do list will be created after the research phase is done. The to do list will be created by the graduate. The to do list will be reviewed and iterated upon by the company

supervisor. The to do list will consist of all the tasks that need to be finished for a successful implementation of the re-identification model.

5.2 Global planning

Seeing the relatively short time of half a year available for the project. A global planning is needed to structure the project and lead it in the right direction.

5.2.1 Responsibility

The graduate will report to the company supervisor the outcomes of different research matters. The graduate will also report his preliminary results or obstacles to the company supervisor. In consultation with the company supervisor small adjustments can be made to the goal of the research or assignment.

The graduate will in a timely manner plan meetings with the supervisor from Saxion to receive feedback on draft documents.

5.2.2 Preparation phase

The preparation phase focuses mostly on preliminary research and experimentation. The existing *deep SORT* re-identification model will be implemented and a small application will be written which goal is to validate that the out of the box model will not work on the apparel constricted scenario. The preparation phase's most important goal is to create insight into the problem and create a better idea of what the solution would look like.

5.2.3 Research Phase

During the research phase there will be a multitude of proof of concepts created. For each model that is being researched a proof of concept will be created which is able to be tested and scored according to the researched evaluation techniques. At the end of the research phase a report will be delivered to Saxion and BallJames, containing the test results of the different models. Kafka will also be extensively explored.

5.2.4 Development Phase

The development phase contains the scalable software solution which can be used to semi automatically extract labeled training data from the kafka topics completely customisable by the user. The multitude of proof of concepts that will be created are also development focused. By creating a sustainable and compartmentalized way of testing and evaluating deep learning models where models can be swapped out easily and test results are clearly visualized, a lot of value and clarity will be added to the assignment.

5.2.5 Conclusion phase

The conclusion phase will be focused around completing the project and the thesis. There will be room for refactoring or restructuring of the code if need be. The conclusion phase will

also be used to discuss certain future work regarding the re-identification model at BallJames and advice regarding this.

5.2.6 Planning

5.2.6.1 global planning

Phase	Start date	End date
0 - Preparation	10/02/2020	01/03/2020
1 - Research	01/03/2020	09/04/2020
2 - Development / implementation	09/04/2020	12/06/2020
3- Conclusion	12/06/2020	03/07/2020

Table 2 - global planning

5.2.6.2 Important dates

Name	Type	Deadline
Plan of approach	Draft Document	01/03/2020
First meeting Saxion supervisor	First Meeting	09/03/2020
Plan of approach	Final Document	15/03/2020
Graduation report	Draft Document	05/06/2020
Graduation report	Final Document	19/06/2020
Reflection	Document	19/06/2020 - 26/06/2020
Evaluation form	Document	19/06/2020 - 26/06/2020
Presentation	Presentation	19/06/2020 - 03/07/2020
Defense	Defense	03/07/2020

Table 3 - Important dates

5.2.6.3 Gantt chart

A domain adaptation of person re-identification with similar apparel

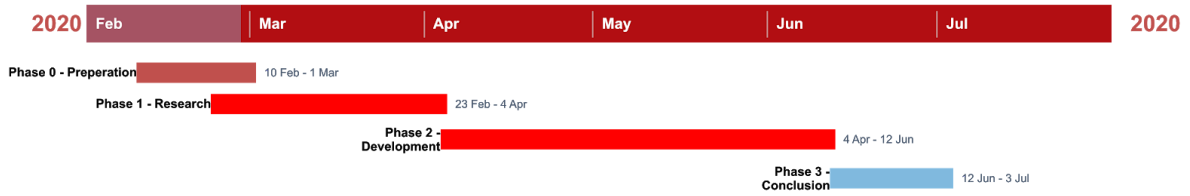

Figure 6 - Gantt chart

5.3 Tooling

The software application will be built in Python upon the SciPy stack which means that everything will be built using NumPy. The data at BallJames is only available in Kafka topics and thus Kafka consumers need to be built to acces this data. Tensorflow is the most used deep learning framework at BallJames and thus will be used during this thesis. OpenCV is one of the most popular computer vision libraries available and will also be used in this project to do image augmentation.

6. Research

6.1 What are the most important currently available solutions to solve person re-identification?

To be able to understand an unfamiliar problem. It's always a good idea to look at the most common techniques that solve such a problem. In the re-identification field there are two important papers which give a decent general idea about the direction a general solution should take. The *deep SORT* paper (Wojke & Nicolai, 2017) gives a general idea about the entire implementation of a person re-identification solution. The *in defence of triplet loss* paper (Hemans & Alexander, 2017) focuses on the appearance descriptor which is needed to mathematically describe a person and later compare it.

6.1.1 Simple Online and Realtime Tracking with a Deep Association Metric

6.1.1.1 Goal

The goal of the *deep SORT* model is to implement appearance information into SORT(Simple Online and Realtime Tracking). Theoretically this would increase the maximum time a person can be occluded and still be recognized as the same person. To ensure the possibility of real time tracking the focus will be on the offline training of a CNN which can generate an appearance descriptor which in turn can be used to combine tracks belonging to the same person.

6.1.1.2 Kalman filtering

Kalman filtering is a tracking technique based on bounding boxes and the possible movement of these boxes based on the history of the track. Kalman filtering can be implemented by following the below mentioned rules.

1. When a new bounding box appears it is classified as tentative. The bounding box needs to be associated(by any kind of measurement such as overlap) to following bounding boxes in the next three frames otherwise the track is deleted.
2. When a track cannot be measurementally linked to a bounding box, bounding boxes will be estimated based on the number of frames where the track is missing in combination with the average movement velocity of the track. These estimations are used as a new basis for new measurement wise associations of new bounding boxes.
3. Estimations will only be done for a specified amount of frames. This amount is commonly referred to as age. When a track exceeds a certain age, the assumption is made that the person has left the frame and the track is removed from the list of tracks.

6.1.1.3 Assignment problem

A common way to link Kalman estimated bounding boxes and newly detected bounding boxes is to create an assignment problem. An assignment problem is a problem which looks to find a one-to-one correspondence between two sets of the same size using a weighted function in such a way that the cost of this newly created set is minimized. The hungarion method is a common way of solving a standard assignment problem. The inner workings of the hungarian method are not of importance in this thesis. The assignment problem that needs to be solved has two important integrated metrics, appearance and motion.

To incorporate motion information into the assignment problem the squared mahalanobis distance is computed between the predicted kalman bounding boxes and the newly detected bounding boxes. The mahalanobis distance is a distance function that keeps the uncertain nature of tracks in account by punishing bounding boxes that are further out from the mean track even though they may be closer in euclidean distance. And rewarding bounding boxes that fall in the general direction the track is heading.

The appearance information is added into the assignment problem by generating an appearance descriptor of a person. This appearance descriptor is generated using an offline trained convolutional neural network. This CNN is trained on the MARS dataset. The appearance descriptors are generated for the last 100 detections in a track. These appearance descriptors are compared to the newly detected people using cosine similarity.

The squared mahalanobis distance between the estimated kalman bounding boxes and the new bounding boxes in combination with the cosine similarity between appearance descriptors of old detections and new detections form the assignment problem which is solved using the hungarian method.

6.1.2 In Defense of the Triplet Loss for Person Re-Identification

The current hypothesis is that an existing model can be used to create useful appearance descriptors when trained on the apparel restricted domain which is the football match. Since there are multiple re-identification models available a selection needs to be made on the most promising model. The embedding referenced below are referenced to as appearance descriptors in the rest of this thesis.

6.1.2.1 Goal

The goal of this paper is to defend the Triplet Loss model. In recent times a classification loss sometimes in combination with a verification loss was often preferred over the Triplet loss model. The paper In Defense of the Triplet Loss for Person Re-identification explains why Triplet loss is more often than not a superior method if implemented correctly and used under the right circumstances.

6.1.2.2 Learning Metric Embeddings

The goal of metric embedding learning is to create a function which can map a person into a metrical space. Ideally this function maps similar people closer in space and unsimilar people further away in this space. This function is most often a simple linear function but can also be complex non-linear and represented by a deep neural network.

6.1.2.3 The Triplet Loss

The triplet loss works as follows. Take 3 images from a list of identities where 1 image is an anchor point which is the image that will be trained upon, the second image is the positive pair which is an image of the same class (person), the third and final image is the negative pair an image of a different class. A set of these 3 images is called a triplet. The embeddings will be computed for each image. The goal of this training is to minimize the distance between the positive pair and the anchor and maximize the distance between the anchor and the negative pair. This loss will be optimised over the whole dataset preferably till all the possible pairs are seen. A big disadvantage of the triplet loss is that with an ever growing dataset, the number of possible pairs grows cubically which makes training on all possible pairs not efficient.

6.1.2.4 The importance of mining

The created function learns relatively quickly to compute useful embeddings for common triplets. A common triplet can be 2 images of the same person wearing the same clothing and 1 image of a person wearing completely different clothing. Intuitively these kinds of triplets are relatively easy to solve. In contrast to triplets that contain 3 images with similar clothing which are a lot harder to compute useful embeddings for. This means that mining of hard positives and hard negatives are important. A hard positive is a positive pair of images which are of the same person in wildly different poses or different angles. A hard negative is a pair of images which contain 2 different people who are quite similar in appearance, clothing and pose. These hard negatives and positives helps the function understand what it “means” to be the same person which is crucial to person re-identification.

6.1.2.5 Common caveats

A common caveat is to mine only hard negatives and hard positives over the entire training set. Which has a negative impact on training since the triplets will be too hard and the function will not start to converge in a timely manner. The paper proposes a modification to the common way of creating triplets. The general idea behind this proposed modification is to form batches from random identities and then sampling random images from these identities. This process results in a batch of random identities and images. The hard mining will only be done inside of these formed batches which in turn results in fluctuating hard to moderate *hard* positives and negatives. This adaptation of the common implementation of the Triplet Loss model ensures a useful function which can describe a person accurately and results in an embedding which is close to embeddings of the same person and far away from embeddings of a different person in a certain feature space.

6.2 How is training data structured in existing re-identification datasets and how should this be implemented for our problem?

Data is one of the most if not the most important part of training a deep learning model. The data for the re-identification model should correctly represent the data that will be fed into the model in production. The model should also be able to learn the nuances which distinguish players of the same team. Since the structure of the data and the way data is presented to the model is crucial in the convergence of a deep learning network. Seeing the criticality of the data research and experiments need to be done to come up with the best way to structure the training data of BallJames.

6.2.1 Existing datasets

There are two most commonly used person re-identification datasets. The Market-1501 (Zheng, Liang and Shen et al; 2015) dataset is the most popular one. It consists of 32.668 annotated bounding boxes divided over 1.501 identities. The dataset contains images from a supermarket close to Tsinghua University. The MARS dataset (Zheng, Liang & Bie et al; 2016) is an extension to the Market-1501 dataset and contains 1.191.003 annotated bounding boxes divided over 1.261 identities.

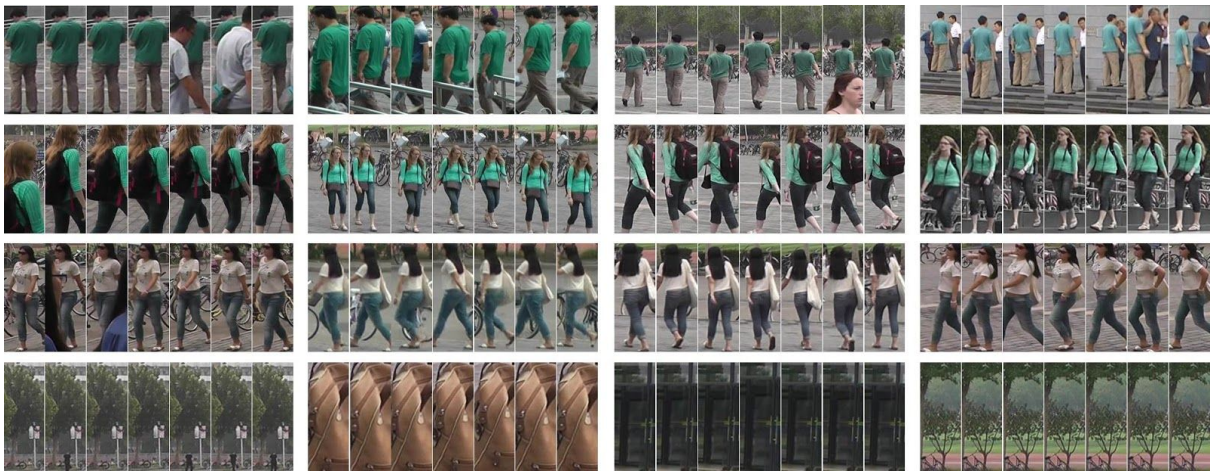

Figure 7 Example of some images in the dataset specific to certain identities

Person re-identification datasets are often structured in the same way. The datasets contain two folders bbox_train and bbox_test. The train folder contains identities that should be used for training a deep learning person re-identification model. The test folder should be used to test the quality of the training and should never be used for training and testing.

The naming convention often includes an id for a certain identity and the location of the bounding box. The frame id is also included in the name of a certain image. This information helps to generalize the dataset for different purposes. Sequential information can be implemented by using the frame id to determine the order that images appear in. The location information could be used to exclude options that could not possibly be the right person seeing that a person can only cross a certain distance in a number of frames. The

location data could also be used when implementing the aforementioned *deep SORT* model and specifically Kalmar Filtering.

The analysis of the two datasets revealed a list of requirements for the BallJames dataset.

- Different sets for training and testing/validating
- Save meta data (location, frameID) in the name of an image
- Structure images by identity.

6.2.2 BallJames dataset

The BallJames dataset should be split in two datasets, one for training the model and one for testing and/or validating the model. The testing part should not be used for training and ideally consist of teams that the model hasn't seen while training to test if the model can generalize on different teams.

The images in the dataset should be stored in different folders corresponding to the identity the images belong to. This should result in a folder which contains a single folder for each identity, filled with images of only that identity.

The names of the images should contain metadata of that specific image. This metadata should extend the range of situations the dataset could be used in. Location data, sequential data(frameId) and identity data(team, number) should be included in the name of the image.

6.3 Can we rely on existing multi view tracking data to generate partially labeled tracking data in a time feasible way?

Training a deep learning model and specifically a re-identification model requires a lot of data. This data can not be collected in an acceptable timely manner by hand and thus needs to be automated. BallJames has possession of enough data to train a model. The challenge is collecting the images needed for training and using the metadata describing the identity of the player to match images to the corresponding identity and thus creating labeled tracking data which in turn can be used to train a re-identification model. The program needed for this type of data extraction can also be used for different applications that need labeled training data and thus needs to be written as configurable and open ended as possible.

6.3.1 Multi view tracking

Multi view tracking is thoroughly explained in chapter 3. For the sake of context it will be summarized here. Multi view tracking is the third step in the BallJames pipeline. The multiview tracker combines the single view tracks for each camera into one track per player. An example of a single frame in a multi view track is pictured below in figure 8. The figure clearly shows player number 21 from Ajax from the 14 different camera angles. The improvement a multi camera system makes compared to a single camera system is also clear. Camera 1 to 8 and camera 14 can't seem to detect the jersey number. Camera 9 to 13 make up for this lack of correct number classification.

Figure 8 - A single frame in a multi view track

The assumption is that if the tracking quality is high enough, the multi view tracker could supply reliable accurate data. That could be used in the creation of the dataset. To verify this

assumption a large sample of the data will be collected and checked by hand to verify that the data does not contain wrongly labeled data. The data that was sampled does not contain any wrongly labeled data and can thus be used for training the person re-identification model.

The multi view tracker greatly improves the quality of the tracking data in comparison to a single view track. It improves the quantity of accurate number and team classifications as mentioned before and shown in figure 9 & 10. The multi view tracker also solves some problems regarding occlusions. A standard occlusion is shown below in figure 9 and b these images are taken from a single camera. This single camera only manages to capture the front of the player when he's crossing behind another player. Intuitively a camera that has a few of the backside of this player is not impacted by this occlusion since the player never leaves the few of the camera and should be detected during the "occlusion".

Figure 9 - *An example of a single view track that is broken because of an occlusion*

Figure 10 - *An example of a single view track shortly after it was broken*

All the data is available online and easily queryable by using Kafka. Kafka allows for easy acces and real time processing which greatly reduces the time it takes to download the images and preprocess the dataset. Another advantage of this real time processing is that it allows for the opportunity to structure the dataset in realtime and removes the need for creating a raw set which would later be processed into the real set thus saving time and money.

Seeing the acceptable accurately labeled tracking data and ease of acces to this data, it is safe to conclude that the multi view tracker supplies a reliable source of data which is easily accessible and processable in a time feasible way.

6.4 Since we're in an apparel constricted scenario we can't rely on a standard global appearance descriptor, will domain specific training data solve this?

The assumption is that global appearance descriptors will not be able to differentiate between players of the same team since these global appearance descriptors are generated by a model which is trained on the Market dataset which consist of people wearing different apparel and thus could use the apparel to distinguish between different people.

An experiment needs to be run to prove this assumption. The triplet loss person re-identification model will be trained on a subset of the dataset. This subset contains 70 identities with 70% separated for training and 30% for testing and validating. The goal of this experiment is to prove that the model converges on the dataset. This would indicate that the model is able to distinguish between the different identities.

Multiple experiments will be run with different hyperparameters to see what parameters help the model converge faster or better. If the model converges it will be tested using the "in the wild" testing program. The results from the testing program should indicate that the similarity scores between players in the same team are lower than the results from the baseline model.

The results will be reviewed in the table below. The newly trained model will be compared to the global appearance descriptor supplied by the *deep SORT* repository. There will be three important metrics that will be compared. The metrics will be summed up below and under each metric there will be explained why these metrics were chosen.

- Same identity to same identity (ground truth)
    - This is the ground truth and shows how well the model learned what a "person" is
- Player in a team to another player in the same team
    - This is the hardest and most important metric, it indicates how well it can distinguish between two players of the same team. This is important when linking tracks together.
- Player in a team to a player in another team
    - This should be the easiest metric since there are a lot of distinct differences between two players of different teams.

	Deep Sort	Triplet loss model
Ground truth	0.9542959786	0.9881430114
Same team	0.8736280714	0.7802436314
Different team	0.6335646043	0.5698762257

Table 4 - experiment results deep SORT vs triplet loss

The results in the table above clearly show an improvement on every metric. Seeing as to how the same team metric has improved by almost 10%, it is safe to say that domain specific training data does indeed supply a more useful appearance descriptor. Which is able to distinguish accurately and reliably between players of the same team.

6.5 What distance metrics should be used when testing the deep learning models?

The purpose of a distance metric in BallJames's domain is to determine the similarity between two appearance descriptors. Ideally this should indicate that when shown two appearance descriptors of the same player they are highly similar and there should be a low similarity when two appearance descriptors of different players are shown.

Distance metrics generally try to plot an appearance descriptor in a high dimensional space. Each distance metric does this a little bit differently. All the distance metrics try to determine their own definition of distance on the data. Some try to determine the literal distance between these points in the high dimensional space, others try to show if a certain data point is in line with the general direction of a group of data points.

The three most commonly used distance metrics are the cosine similarity, euclidean distance and the mahalanobis distance. The mahalanobis distance tries to show if a certain data point is in line with the rest of the data. This is different to the euclidean distance which simply checks the "how the crow flies" distance between two points. In the example shown in figure 11 it shows the general data points in a kind of diagonal oval. A new data point is surrounded by a red square. The euclidean distance between the average of all the data points and the new point is smaller than the mahalanobis distance in this case, however the mahalanobis distance tries to adjust for the direction the data is heading in and use this knowledge to predict the chance that a new data point belongs to the group of other data points.

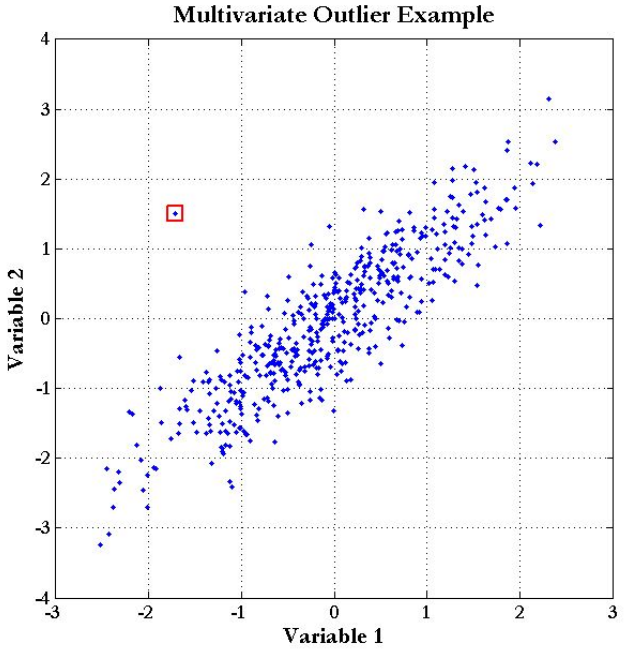

Figure 11 - Mahalanobis distance example

The cosine similarity takes a different angle to computing the similarity between two values. Cosine similarity plots two or multiple data points in a high dimensional space and computes the cosine of the angle between these points. A big advantage of this technique is that since it only looks at the angle between points the values or the size of the values does not matter since this only increases the euclidean distance and not the angle between these points.

Seeing as to how the *deep SORT* paper (Wojke & Nicolai, 2017) proposes to use the cosine similarity, this will also be used in the BallJames implementation. Since the size of the appearance descriptor values do not matter and only the difference between these values are of importance, cosine similarity will be used.

6.6 What evaluation methods are important when testing a deep learning model for this specific use case?

To be able to determine the successfulness and the convergence of a re-identification model during training it is important to use the correct evaluation methods. Since these evaluation methods can be unique for a re-identification model. Insight is needed to pick the correct evaluation method and what its results mean.

There are two main steps regarding validation of the model. The goal of validation is to get an indication of the performance of the model. Just because a model achieves a low loss and a high accuracy during training does not mean that the model will perform well on unknown data and a production environment. For this reason a model is validated during and/or after training.

Step one is to validate the model during training. The main goal of validating during training is to see if the model performs well on data it hasn't seen yet, it should show a high accuracy on the validating data. A validation set should contain data that is not known to the model and thus not present in the training set.

The second step is to create an “in the wild” testing environment. This environment should be as close to the production environment and ideally show good performance. This in the wild testing program is a simple implementation of the final production version. In the wild testing should only be done once a promising model is reached.

For BallJames the first step was already implemented in the training code base. The second step is built in python and uses a raw version of the dataset. The data is fed into the program from an offline validation set. The validation set includes identities of teams that the model has seen and completely new teams. The identities of a team that the model has seen are added to see how well the embedding is computed. The unknown team identities are added to test the generalisation of the model.

7. Design

The design of the data generator has to have some important characteristics. It needs to be scalable to different camera setups. Seeing as to how the data generator could be used in the future at BallJames, it should generate a raw dataset which could later be used to generate a final dataset.

7.1 Architecture

The architecture of the data generator is as follows. There are five main components.

- Consumers
- Data pipeline
- Data extractor
- Kafka
- Config file

The consumers query the data from kafka. The data pipeline synchronises all the different consumers since these can start from different offsets(frames). The data extractor reads all the data from the different queues which are created in the data pipeline and exports the data in a specified folder structure. The config file specifies the address and the topic names of the data, any specific consumer settings, the amount of cameras with their specific location next to the pitch and the specific start time of the match.

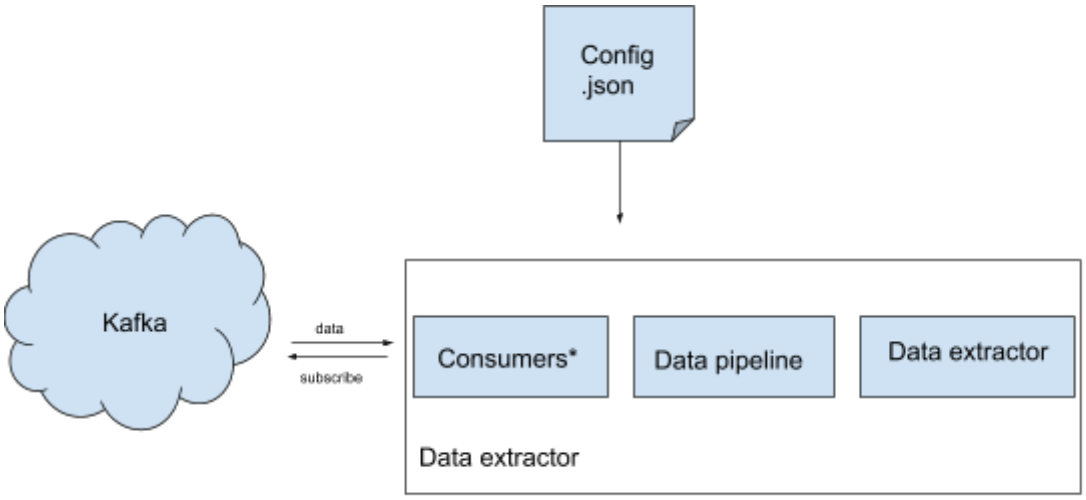

Figure 12 - Data extractor architecture

7.2 Scalability

Scalability is one of the most important characteristics of the data generator, seeing as to how there could be a different number of cameras that were used to record a match. The components of the BallJames pipeline are changed and improved constantly. This means that there could be new topics introduced at a later time which should be available for

downloading with the data generator. To ensure this possibility a config file is used. This config file follows the BallJames config file setup to ensure code quality.

7.3 Abstractness

Abstractness is another important quality of the data extractor. There will be different use cases in the future at BallJames where the data extractor could be used to generate labeled training data for. To supply this kind of abstractness, the data will be exported in a raw dataset form which can be shaped into any dataset that is needed for the problem ahead. The shaping of the raw dataset can be done by using the two supplied dictionaries. These dictionaries describe the data and its structure. The dictionaries map the following data.

Dict 1: CameraID -> frameID -> Single View Track ID -> Batch Handler Track ID

Dict 2: CameraID -> Batch Handler Track ID -> Single View Track ID -> detection(Identity information)

The information supplied by the dictionaries should be sufficient to create datasets for a large variety of problems. For the dataset used in the training of the person re-identification model a script was created which extracted all the single view tracks in a certain multi view track and combined these into one identity. Also use the identity information to name the image with the correct location, identity and frame id.

8. Implementation

8.1 Training data generator

There are a few things that need to be accounted for in the training data generator. There needs to be a balance between time efficient downloading and not filling up the memory too much. The training data generator starts by reading in the config file which is structured in the BallJames config file way. The config file is in JSON format and contains information about the match and the Kafka topics. The parsed config file is passed onto the data pipeline.

8.1.1 Multiprocessing queues

The data pipeline instantiates two groups of multiprocessing queues which correspond to the two topics(Batch handler, detections). The batch handler topic contains information about the track of a player and its identity. The detections topic contains bounding boxes of all the detected players in a certain frame by a certain camera. A single message from the detection topic also contains a base64 encoded image of the bounding box. The job of the training data generator is to link the identity and track information that is inside of the batch handler messages to the detection images which are in the detection messages and output these files into a certain structure.

The multiprocessing queues are used to ensure a certain synchronicity between the different consumers. Seeing as to how the base64 encoded 4k images are quite large it's not possible to keep filling up queues and emptying them at the end, so there is a need for a synchronized state between the batch handler and the detections topic to be able to empty the queues at the same time. The queues are emptied ones every consumer has filled up its corresponding queue. The queues are also emptied once the batch handler messages are ahead frame id wise to the detection messages seeing as to how the consumers only consume messages in chronological order.

8.1.2 Matching messages

Matching messages is a big part of the generation of training data in BallJames's case. This happens in the data extractor. This matching needs to happen because there are two different messages that contain data that is important for the data collection. The batch handler contains messages that have information regarding which detections belong to which player and track. The detection messages contain the images corresponding to the detections. These need to be matched to know which images need to be extracted.

A while loop runs until all the queues are filled up. When all the queues are full, firstly the batch handler queue is emptied and added into a dictionary with as key the frame id which is very important when it comes to matching the messages. Secondly the detection queues are

emptied into a second dictionary. The second dictionary is a nested dictionary with the outer dictionary keyed by camera id and the inner dictionary keyed by frame id.

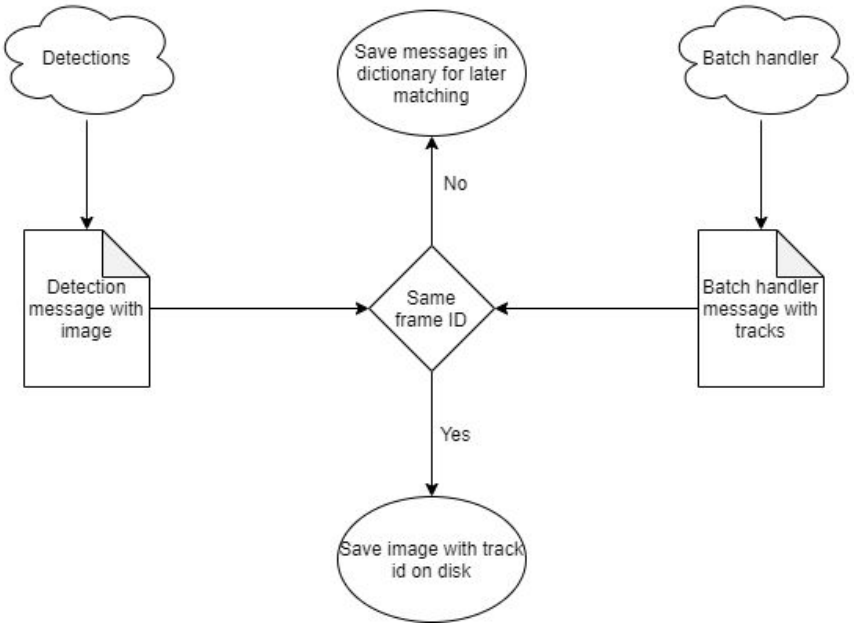

Figure 13 - A diagram showing what needs to be done with the messages

The matching happens after both the queues are emptied. Firstly the batch handler keys are sorted ascendingly. This newly created list is looped over by index. The index looping happens to be able to loop through both the dictionaries from one for loop. At the start of each loop the frame id key corresponding to the index is retrieved from the sorted key list. Then the batch handler track object is checked to see if it contains any batch handler tracks. This needs to be done since there is a batch handler track for every frame even if there are no players on the pitch which would result in a batch handler track object without any batch handler tracks.

Inside a batch handler object are ideally multiple batch handler tracks containing multiple cameras that detect a player. The tracks in a batch handler object are looped over to be able to know what images from a detection message need to be extracted. This loop extracts the following information from a batch handler track.

- **Single view track id**, which is saved inside of a batch handler message since the batch handler message originates from this single view track.
- **Real world coords**, which could be used for a more precise distance metric between players.
- **Camera ids**, the camera id is used to query the right inner detection dictionary.
- **Bounding boxes**, the bounding boxes are used to match an image from a detection message to the detection without an image that is in the batch handler.
- **Types**, the type is either a player, referee or unknown; this could be used to filter out the referees or not correctly detected players.

- **Confidences**, the confidences lend the opportunity to filter out detections that are uncertain to great a theoretically more accurate dataset.

A batch handler track contains a number of detections that need to be matched. These detections are looped through and checked to see if there is a detection message with an image available for this batch handler track detection. A for loop is created that loops over the detections in a detection message when a detection message is found for this batch handler track detection. In this loop the bounding boxes of both detection messages are compared, if a match is found the global dictionary that maps a batch handler track id to the player identity is updated, the image that is in the detection message that corresponds to batch handler detection is saved in the raw folder structure and a tuple containing information about the match between the two objects is saved in the matched objects array. This array is returned and used later to update the global dictionaries and to delete unmatchable and already matched messages from the detection dictionary and the batch handler track dictionary.

8.1.3 Deleting messages

Deleting messages is an important part of the training data generator seeing as to how this ensures a low memory usage. There are a few instances where messages are deleted:

- Matched messages, messages that are already matched and saved to disk should not be kept in memory.
- Unmatchable messages, messages that do not contain batch handler tracks should be deleted and their detection message counter part should too, detection messages that have a frame id lower than the lowest batch handler track message should also be deleted seeing as to how these will never appear and thus are not matchable.

8.2 Triplet loss model

The triplet loss model's implementation is provided by the writers of the triplet loss paper. Since this implementation is used during training of BallJames's triplet loss model the implementation will be explained conceptually and not in detail.

8.2.1 Training

The training process of the triplet loss model is quite similar to other deep learning models. The dataset is split into two. One part for training and one part for validation/testing. There needs to be a balance between the sizes of the two parts. The validation set does not improve the model and thus not too much data should be wasted on the validation set but the validation set should contain all the core principles that are represented in the dataset. This ensures that the results of the validation properly represent the production scenarios. The training set is arguably the more important part of the dataset and should be set up in such a way that the set lends the opportunity for the model to learn what needs to be learned to achieve the goal of the model.

At the start of training the entire training set is randomly shuffled. After the shuffle the training set is divided into batches as the paper recommends. Batches are simply a small

randomly sampled part of the training set which contain a specified number of identities and a certain number of images per identity. The amount of identities and images can only be determined by experimenting with different numbers and analyzing the results. The model is shown the multiple batches per epoch. An epoch is one iteration of all the training data. After an epoch has finished the data is once again randomly shuffled and again divided into batches. At BallJames four identities and five images per identity are shown per batch.

The batch data is then divided into positive and negative pairs with an anchor. The anchor is the image that an accurate appearance descriptor should be generated for. The positive pair is the anchor in combination with another image of the same identity, ideally this other image is an image that is the least similar to the anchor which should make it hard to recognize as the same identity. The negative pair is the anchor in combination with an image from another identity and ideally an image that looks highly similar to the anchor which would make it hard to differentiate. This process is called mining and should give the model the tools to learn what it “means” to be a person instead of recognizing a person by its clothes or pose.

A small modification to the triplet loss model’s standard implementation is made to try to make the model better understand what an occlusion looks like. The adjustment is that only images in a sequence are shown to the model for every identity which means that instead of the standard eighteen randomly sampled images per identity, the model is shown eighteen images of a sequence. This is made possible by the meta data that is encoded into the name of an image especially the frame id. This process continues for a certain amount of steps but should be stopped when the model reaches a loss that does not go down any further. The loss is generally a metric which explains how well the model can predict data and shows how far off the model is. In the triplet loss model the loss shows the distance between the positive pair’s and the negative pair’s appearance descriptors.

8.2.2 Validating

Validation is done after the model has finished training and a few times during training. At BallJames the model is validated every 5000 steps. This validation during training can be used to see if the model is overfitting or generally improving. If the loss is going down but the validation accuracy does not increase or even gets worse it means that the model is overfitting on the training data and not generalising well.

A validation script will run after the model has finished training and the results look promising. This validation script aims to simulate the production environment to correctly show how well the model is performing. The validation script shows the model different images of different players in the same way the eventual production implementation will do. These images will be encoded into an appearance descriptor by the model and then compared to each other. The appearance descriptors are compared using cosine similarity. Ideally images of the same player have a high similarity and images of two different players have a low similarity.

9. Experimentation and results

There are three metrics that are important to judge the result of BallJames's. These metrics were explored in the sixth research question but will be repeated here for the sake of clarity. In BallJames's use case there are a few different scenarios which need to be handled to match broken tracks together. A broken track could be compared to a single view track of the same player, of a different player on the same team and to a player of another team. These three scenarios translate into the three defined metrics:

- Ground truth
    - Player to different image of the same player
    - Ideal scenario: 1.0
- Same team
    - Player to different player of the same team
    - Ideal scenario: 0.0
- Different team
    - Player to player of a different team
    - Ideal scenario: 0.0

These metrics should give an accurate indication how well the newly trained model will perform in the production environment. These metrics will be shown in tables when discussed in the results. The value in the table is a value between 0.0 and 1.0 and indicates the average similarity in the validation set for that specific metric.

The dataset is an important part of the training of a deep learning model but not the only deciding factor. Hyperparameters are another impactful factor in training. The hyperparameters dictate the amount of identities and the amount of images shown per batch. The hyperparameters also set the number of trainingsteps, how often validation needs to be done and if any preprocessing is needed on the images.

There is often not one generally ideal setting for each hyperparameter. The only way to find the best hyperparameters is by experimenting with different numbers and analysing the results. Experimentation can also be done on different datasets which could prove the potential of the model by letting it converge on a limited version of the dataset.

The successfulness of the training can be determined by looking at the loss and the precision. The loss is a value which represents how far the model's predictions are from the truth. The precision displays the amount of correct predictions or in BallJames's case correct generations of an appearance descriptor, divided by the number of all returned results.

In the triplet loss model the loss is a value which represents the difference between the distance of the positive and negative pair. If the distance between the appearance descriptors of a positive pair is smaller than the distance between the negative pair then the loss will be lower seeing as to how these are the preferred results.

For BallJames around 50 training experiments were done. These experiments use different hyperparameters and different training sets. The results and details of the experiments will be explained below. The three biggest experiments will be discussed for the sake of compactness. The other experiments were mainly focused on hyperparameters and impacted the graphs minimally.

9.1 Datasets and hyperparameters

There are three main datasets used during experimentation. First it is important to see if a model could converge on the newly generated dataset. A small subset of the big dataset is created to test this. Ideally the graphs show that the model can overfit on this small subset. When this is shown in the graphs it proves that the model is able to learn the data and gives the go ahead to try a larger dataset and properly train the model on that.

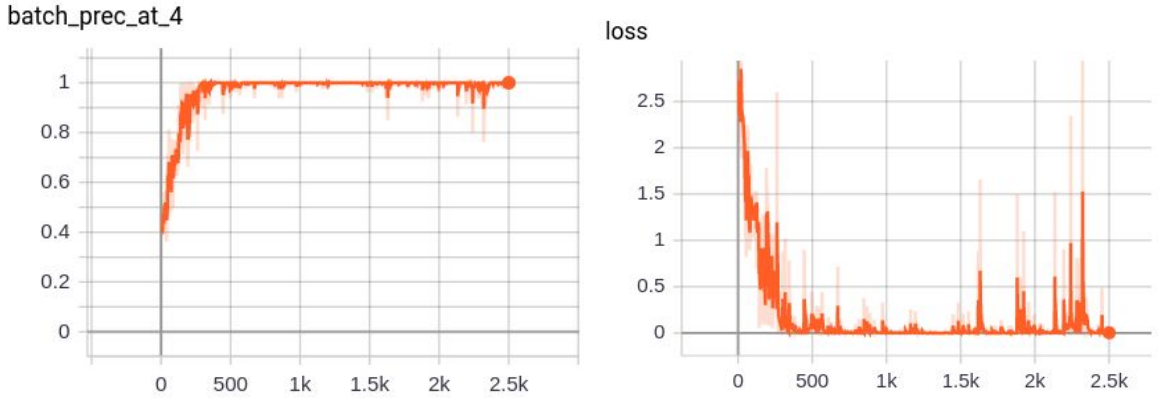

Figure 14 - Precision and loss from the limited dataset

In figure 14 it is shown that the precision reaches 1 after around 500 steps which is expected on a small dataset but it shows the model's ability to converge. This ability to converge on the data proves that the model is able to learn the data and this is important for learning the data. The loss of the first training experiment reaches 0 after around 500 steps which is fine. The spikiness of both graphs is due to the mining of hard positives and negatives. For each batch the difficulty of these pairs differentiate and thus could lead to slightly worse results. Seeing the limited size of this dataset it will not be validated using the in the wild testing validation program.

After the go ahead given by the limited dataset, the second training experiment will be on a unique dataset. The unique dataset contains around 70 identities with the same 70/30% training/testing split.

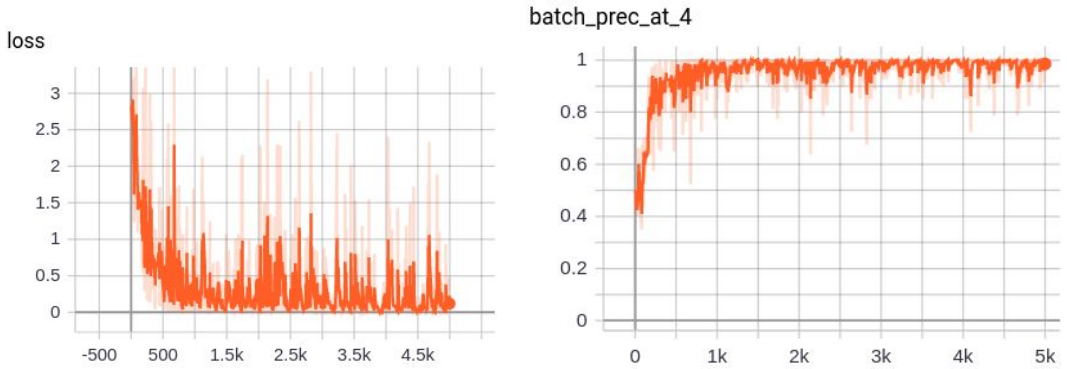

Figure 15 - Precision and loss from the unique dataset

The graphs show that the model is able to converge on a bigger dataset. The dataset contains only one player per identity and that player does not appear in a later different match or a later moment in the match under a different identity. This should give the model the ideal situation to converge on a bigger dataset. However the limited size of the dataset could hurt the generalization of the model seeing as to how the model does not train on a lot of data. The loss averages out close to 0 at around 1.5k steps. The precision averages out around 1 at the same amount of steps. This shows that the hyperparameters of 4 identities and 5 images per identity per batch are working and do not need to be improved.

These are the results of the validation script on the unique dataset showing a big improvement in the same team metric which is the metric that is most important for BallJames's use case.

	Deep Sort	Triplet loss model unique dataset
Ground truth	0.9542959786	0.9881430114
Same team	0.8736280714	0.7802436314
Different team	0.6335646043	0.5698762257

Table 5 - Validation results unique dataset

The last big training experiment involves the final big dataset with in total 1300 identities. These identities use the same 70/30 split regarding training and testing. This constitutes to around 900 training identities and 400 testing identities. Players appear multiple times over different identities from different moments in the match. Players over multiple identities are added to try to give the model an understanding of short term re-identification. The goal of this is to learn the model that the real identity of a player doesn't matter but that when given images of the same player it recognizes the two images as the same player. This should give the model the ability to re-identify a player after an occlusion occurs.

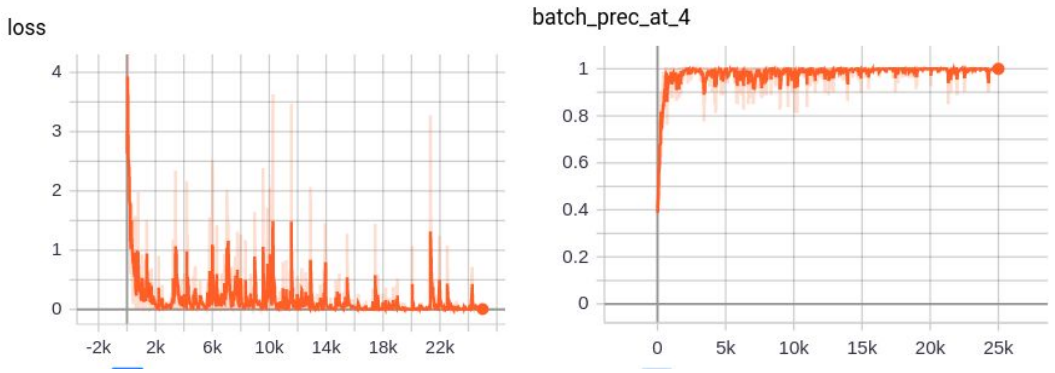

Figure 16 - Precision and loss of the big dataset

Figure 16 shows quite a spiky graph. This might be due to the players appearing in multiple identities. The model might have learned that a certain player belongs to one identity, however at a later time the model is shown the same player in a different time in the match belonging to a different identity. This might be the cause of the spikiness in combination with the mining issue which was discussed before.

The results of the in the wild testing are displayed below in table 9.2.

	Deep Sort	Triplet loss model unique dataset	Triplet loss model big dataset
Ground truth	0.9542959786	0.9881430114	0.986075352
Same team	0.8736280714	0.7802436314	0.637050754
Different team	0.6335646043	0.5698762257	

Table 6 - Validation results big dataset

Table 6 shows that the big dataset makes another improvement in the same team metric. The metric is at a level which should give BallJames the tools to combine broken single view tracks seeing as to how the average similarity between the identities from the same team are around 0.60 which should be a big enough difference to make up for the different poses and illumination that players could have after an occlusion. The different team metric is not tested for the final training since this metric is good enough in all the models and this data could now be used for further training which should increase the generalisation of the model.

10. Conclusion

Person re-identification cannot be performed with the standard global appearance descriptor supplied by the *deep SORT* paper(Wojke & Nicolai, 2017). The *deep SORT* appearance descriptors of players from the same team are too similar which makes it impossible to reliably differentiate between them. Differences in pose and illumination could make a different player of the same team with the same pose and illumination as a player before an occlusion more similar to the track before an occlusion than the correct player.

This thesis proves that BallJames's appearance description generator generates appearance descriptors which can accurately differentiate between players of the same team. This helps BallJames in combining tracks that are broken due to an occlusion. Which in turn would improve the accuracy of the tracking data. The results discussed in the previous chapter show that person re-identification can be performed on people who are similar in appearance when the appearance descriptor generator is trained on domain specific data. This has not been done before and thus adds a lot of value to the person re-identification field and BallJames as a company.

The appearance descriptor generator can easily be implemented by following the next steps.

1. First generate an appearance descriptor of the last twenty frames of a broken track.
2. Generate appearance descriptors of a newly found track of a player.
3. Compare the two groups of appearance descriptors using cosine similarity.
    a. If the similarity score is above a certain threshold the two tracks belong together and should be combined
    b. If the similarity score is below a certain number the search should be continued for the continuation of the track.
4. When all players are tested and score below the threshold, the player is probably out of the view of the camera and the search should be ended.

Using these steps and the newly created triplet loss model, BallJames should have the tools to fix broken tracks during a match or after the match as a post processing tool. The triplet loss model can be implemented in the current pipeline of BallJames.

The data extraction tool that was built during this graduation project can and will be reused at BallJames. It is scalable and abstract enough to be used on a range of different projects and thus adds a lot of value for BallJames. Data extraction used to be quite hustle of a lot of different scripts which can only generate data for one purpose or model. With the new tool a standard BallJames config file can be supplied specifying which match and what topics need to be extracted. These specified topics will be downloaded and can be stopped at any time. The data extractor streamlines the entire dataset creation portion of creating a new model which is often the hardest part.

11. Reflection

I am very happy with the result of the thesis. I proved that person re-identification can be implemented at BallJames and has the potential to greatly improve the tracking data. I am really proud of this because there was no existing paper or project where domain specific data greatly increased the performance of the person re-identification model. To add to this person re-identification has not been implemented in an as highly constricted apparel scenario as is a football match. I think because of this my thesis has met all expectations and in some areas might have exceeded expectations.

I think that the data extractor is very easy to use and fits neatly into the BallJames environment seeing how it reuses the standard BallJames config files and is set up in such a way that it is easily customizable for any purpose.

The kanban process helped guide my development in the right direction. The flexible nature of the kanban process proved vital in the rapidly changing environment surrounding my project. Especially the output of the data extractor was changed multiple times after consultations with different co-workers and after analysing the existing datasets.

My time at BallJames is probably the best time I've had since starting HBO-ICT software engineering at Saxion. The environment at BallJames stimulates learning and doesn’t punish mistakes. Being surrounded by experts in artificial intelligence taught me a lot about the process surrounding artificial intelligence and how artificial intelligence can be used in a commercial product.

The biggest obstacle I faced during this thesis is the corona situation. I had to work from home to comply with the guidelines set by BallJames. This meant that I could not discuss my project face to face with my supervisor. This slowed my progress quite significantly seeing the difficult nature of the assignment and the importance of being able to ask simple questions on the go.

12. References

Hermans, A., Beyer, L., & Leibe, B. (2017). In Defense of the Triplet Loss for Person Re-Identification. *arXiv preprint arXiv:1703.07737*. Retrieved from https://arxiv.org/pdf/1703.07737.pdf

Wojke, N., Bewley, A., & Paulus, D. (2017, September). Simple online and realtime tracking with a deep association metric. *2017 IEEE international conference on image processing (ICIP),* 3645-3649. IEEE. Retrieved from https://arxiv.org/pdf/1703.07402.pdf

Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., & Tian, Q. (2015). Scalable person re-identification: A benchmark. *Proceedings of the IEEE international conference on computer vision,* 1116-1124. Retrieved from https://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Zheng_Scalable_Person_Re-Identification_ICCV_2015_paper.pdf

Zheng, L., Bie, Z., Sun, Y., Wang, J., Su, C., Wang, S., & Tian, Q. (2016). Mars: A video benchmark for large-scale person re-identification. *European Conference on Computer Vision*, 868-884. 10.1007/978-3-319-46466-4 52.

Research Methods. (2018, February 9). Retrieved from http://ictresearchmethods.nl/Methods

Kreps, J., Narkhede, N., & Rao, J. (2011). Kafka: A distributed messaging system for log processing. *Proceedings of the NetDB*, *11*, 1-7. Retrieved from http://pages.cs.wisc.edu/~akella/CS744/F17/838-CloudPapers/Kafka.pdf