UNIVERSITY OF BUCHAREST

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

MASTER THESIS

SCIENCE COORDINATOR

LECT. DR. PADURARU CIPRIAN IONUȚ

STUDENT

POESINA EDUARD-GABRIEL

BUCHAREST

SEPTEMBER 2021

UNIVERSITY OF BUCHAREST

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

MASTER THESIS

NATURAL BEHAVIOUR PREDICTOR

SCIENCE COORDINATOR

LECT. DR. PADURARU CIPRIAN IONUȚ

STUDENT

POESINA EDUARD-GABRIEL

BUCHAREST

SEPTEMBER 2021

Motivation

The most primeval of philosophical thoughts concerning the meaning of life challenge how we percept the world as rational and sentient beings. What does it mean to be human? This question has been tackled numerous times by erudites ranging from the ancient western philosophers to the more recent of philosophical minds such as Sartre, Merleau-Ponty and Emanuel Levinas. One of the earliest distinctions between humans and animals was noted by Aristotle. His Weltanschauung (personal beliefs regarding world and life) was centered on the idea that humans are rational beings driven by their simple and primal instincts similarly to animals; In contradiction to his aphorism, modern studies seem to prove that animals also possess rational thinking; Novel nonlinguistic models were constructed to comprehend the though process of advanced mammals. Out of the latest models, of great impact is the model proposed by Cameron Buckner, published in the article "Rational Inference: The lowest bound"; The study followed the complex behavior of elephants, chimpanzees, lions and many other advanced mammals; The animals proved an "executive control" which implied that given a goal, the animals consider various complex possibilities to fulfill it before taking an action.[1]

If rational thinking can be found in animals, then what makes us different? A more modern approach in differentiating humans and animals is based on our needs. Abraham Maslow publishes in 1943 his work "A theory of human motivation" in which he proposes a hierarchical structure for human needs based on their importance.[2] Maslow notes that physiological needs proceed any of other needs: safeness, belongness, esteem and self-actualization. Uriel Abulof contradicts his theory stating that the first few levels of needs are common among all anthropomorphic life and that self-actualization is not fully and correctly defined; What does improving one's self mean? Even animals tend to be the alpha male of the group; Could this be our animalistic equivalent?

The four basic needs are strongly correlated with social behavior; We have observed social interactions in a great range of species, from the most primitive (such as microbes) [3] to the most advanced. [4] If animals also do present social behavior, then how do we really differ? The complex and intrigant answer of this question can be offered only after our understanding of social behavior

is complete; This is one of the most difficult tasks at hand. Understanding our social nature and the complex and hidden interactions that we cannot observe easily;

Motivated by this difficulty I decided to dedicate this work towards improving our knowledge in the domain of social behavior understanding. The scope of this master thesis is to provide a deep study of cornerstone approaches, novel techniques and also reinterpret our understanding of human social behavior by developing a custom software which seeks to discovering the truth behind our unseen and complex social interactions;

Table of Contents

1. Introduction

1.1 Historical Nuances

The start of the era of computers can be pinpointed to the creation of the first transistor by John Bardeen and Walter Brattain in year 1947, which was later improved with a more robust design by William Shockley in 1951. The invention brought world-wide acknowledgement as they have been awarded the Nobel Prize in Physics. Marking yet another successful step into digitalization was the development of the first microprocessor: the four-bit microprocessor Intel 4004. The design was easily superseded in the following months and computers by microprocessors which would allow 8-bit operations. From that point on the production of microchips has started to follow Moore's prediction (the number of transistors in a microchip is doubled every two years);

But computers as any other novelty is met with skepticism at first. Given the low rate of trust in computer at the start of the digitalization era every automated process was supervised by humans even at the lower levels; As we have increased our engineering skill, computers started to be less and less prone to error and needed less supervising from humans. After so much progress we have reached a point in which computers are the ones that analyze and signal the apparition of human error;

The digitalization era was not only marked by the development of computer devices but also the birth of new data generation processes; Data is currently one of the few resources we are not in any close do depleting; In fact, IDC stated in the paper "The digitalization of world. From Edge to Core" published in 2018 that the total quantity of data available is forecasted to reach 175 zettabytes in 2015. A quantity so high that we cannot even fathom to employ humans to manually analyze the data. The only scalable solution is allowing computers to parse, represent and event interpret it the information around us.

Originally, we have employed computers to solve problems that require the completion of easy tasks which can be reduced to simple calculus. But this approach can only be taken in very specific situations. How are computers then supposed to solve difficult tasks which we cannot

mold as previously described? The most natural solution would come if computers could think for themselves and propose ways of approaching a problem on their own. In other words, we seek to instill Artificial Intelligence into our computers;

The start of machine learning is marked by three major meetings: “Session on Learning Machines” (1955), “Summer Research Project on Artificial Intelligence”, held by John McCarty the symposium “Mechanization of Thought Process”.. In the chairman introduction of “Session on Learning Machines”, Willis Ware states that the purpose of the meeting was not to convince researchers that future learning machines should be developed for general-purpose but rather that machines are a highly versatile and robust tools which can accommodate many problems and in time, with the improvement of the computing power of machines their ability to accommodate to more difficult tasks will scale up. The term Artificial Intelligence was firstly noted by John McCarty in the year 1956 in an organized workshop at Darmouth College. The proposal of the workshop was to create a system that could capture every aspect of intelligence, what we usually call general intelligence.[4]

Some of the early successes in the domain of AI such as Arthur Samuel’s checkers game software and the Logic Theorist developed by Newell & Simon, a mathematical software which searched possible proofs for theorems based on solution searching and heuristics filled up with ambition the heart of mathematicians and computer scientists. But the domain easily got overhyped and researchers become discouraged by the difficulty of the problems they faced. A famous New York Times article described the efforts of computer scientists to implement a computer translator. The target of the machine was to translate phrases from English to Russian and vice-versa. One of the most famous blunders of the machines where the phrases “The spirit is willing, but the flesh is weak” which was translated as “The vodka is good, but the meat is rotten” and “Out of sight, out of mind” with its amusing equivalent “Invisible, insane”. The article mentions the incredible speed of machines when translating but also discredits their superficial understanding of the words; The word-by-word translation proving that the computers do not comprehend the true meaning of the phrases.[5]

The limited computational power and information of our systems combined with the low number of AI researchers provoked an underwhelming sentiment from the government agencies that redirected their funding into the study of AI; The negligence of AI researcher marked what

would be named as the AI Winter a period of backlash in allocated resources. Researches started to look into knowledge-based reasoning to counter their computation and information limitations and created the first expert systems. Expert systems where machines designed to make use of the knowledge of experts of various domains which would interpret their reasoning method with simple if-then instructions. The scope of their software was limited but compared to the previous results the new expert systems provided solutions that could be used in industry. Some of the more known knowledge based-systems such as DENDRAL (software which infers the molecular structure from mass spectrometry), MYCIN (antibiotics recommendation software) and XCON (a recommender system for computer parts based on client needs) served as proofs of the system's potential. But quickly enough, dead-end started to appear. Deterministic rules can not react properly to the uncertainty of real life and the meticulously written rules became more and more cumbersome to manage and maintain in large-scale projects; As the progress of knowledge-based system staled, the enthusiasm of the public decreased and as a consequence the allocated funds for AI projects were reallocated to other domains, marking the second AI winter.[6]

Another strong tradition for artificial intelligence was neural AI, which had its roots in neuroscience. The base works of neural AI was laid by Frank Rosenblatt. Motivated by the mathematic model proposed by McCulloch and Pitts. The pair's focus was to mimic the impressive processing capability the human mind by imitating the basic human nervous system block: the neuron. They claimed that in essence a neuron is a logic gate and can be easily expressed mathematically. The designed model incorporated the mathematical equivalent of dendrites (as inputs) and axons (as outputs) and the output of the neuron was modeled as a binary result. The "McCulloch-Pitts" neuron could then be attached end to end to other similar unit and compose a intricate networks which could compute arbitrary logical functions; Even though the strength of the model was proven the modelling technique was missing a learning method; Donald Hebb would propose in 1949 a local learning technique (the perceptron algorithm for training single-layer networks) and created an application which would successfully classify images;[7][8]

Algorithm: Perceptron Learning Algorithm

$P \leftarrow inputs \quad with \quad label \quad 1;$
$N \leftarrow inputs \quad with \quad label \quad 0;$
Initialize $\mathbf{w}$ randomly;
while $!convergence$ **do**
  Pick random $\mathbf{x} \in P \cup N$;
  if $\mathbf{x} \in P \quad and \quad \mathbf{w}.\mathbf{x} < 0$ **then**
    $\mathbf{w} = \mathbf{w} + \mathbf{x}$;
  end
  if $\mathbf{x} \in N \quad and \quad \mathbf{w}.\mathbf{x} \geq 0$ **then**
    $\mathbf{w} = \mathbf{w} - \mathbf{x}$;
  end
end
//the algorithm converges when all the inputs are classified correctly

Figure 1. The Perceptron Learning Algorithm

One of the early demises of neural AI research was marked by the publication of the book "Perceptrons" in 1969 by Marvin Minsky and Seymour Papert where the mathematical properties of perceptrons were studied in detail. One of the most important yet trivial results was that non linearly separable data could not be classified through the means of a simple perceptron (the authors proposed the XOR operation as an example); Even though the authors did not discredit the capabilities of multi-layered networks, the launch of the book marked the rise of popularity of symbolic AI over neural AI; The general interest was only to be renewed in the 1980 with some important breakthroughs such as the Neocognitron, popularization of backpropagation for multiple-layer networks and the development of a software by Yann Lecun which recognized handwritten digits on envelopes; The direction which neural AI has taken in the last 20 years is the deep learning approach. The development of AlexNet in 2012 convinced a great portion of computer vision scientists that deep learning will be the dominant paradigm of the domain.

Computer vision represents the ability of computers to bridge the gap between simple pixels and the meaning of the image that the pixels derive from; Understanding the complex and intricate relationships which our eyes innately understand; The human eye is stunningly accurate and adaptive to its environment. Perspective, shape, color, depth, shade are easily captured by our visual prowess; We aim to do the same for computers;

Computer vision started as a project proposed in 1966 by Seymour Papert at the Massachusetts Institute of Technology. The goal of the "Summer Vision Project" was to develop a computer system which could successfully perform three different tasks described by Seymour as "Figure-Ground Analysis" (the distinction between foreground objects and the background) "Region Description" and as a final task "Object identification" (the identification of an object is in a certain image given a known corpus). Sixty years later, many fields benefit from the newest technologies developed in the domain computer vision; A variety of real-world applications made a real impact in various economic branches: Optical Character Recognition – Used in Retail and Postal services, Motion Capture – Used in movies and video games, Surveillance – the analysis of videos and interpreting of human behavior.

Computer vision is a field set to stay in or path of developing, and given the importance of the domain I have decided to dedicate this work into improving own knowledge and systems;

1.2 Problem discussion

The challenge of trajectory prediction is one of the most predominant tasks that appears in the literature of computer vision researches. The ability to accurately predict future positions of objects in various scenes is a technological breakthrough which will lead us closer to autonomous AI as we are in search of understanding the inherent way humans predict what will happen in short intervals of time in a vicinity or how we restrict future to only a few possible outcomes over a larger span of time. Over the plethora of possibilities, we can rationalize and decide which outcomes are more probable and act accordingly. When acting as pedestrians or driving vehicles our brains analyze our environment and we can subconsciously take decisions. In AI one of the governing rules is Moravec's paradox, which states that it is of low difficulty to make computers perform as proficient as adults but it is incredibly difficult if not impossible to make AI have the same comprehension power as a one year old. We are continuously researching methods to allow computers to function as humanely as possible.

1.2 Purpose and research method

The purpose of the current paper is to contribute to the research of trajectory prediction of pedestrian in various scenes. My research efforts have been concentrated in three various directions: analyzing the state-of-the-art techniques used in pedestrian trajectory prediction, collecting data pertinent to training such models and creating a custom-tailored data gathering pipeline which can effortlessly create sufficient information for us to train and validate the employed models and integrate some of the previously referenced models into a simulator. Additionally, I have researched new methods to improve model accuracy of the state-of-the-art techniques by adding new pertinent data streams in the models.

2. Technical Approach

2.1 Technology Stack

We will try to implement a computer vision software which will be able to recognizes movement of people in videos and predicts future positions of humans by observing the various interactions between them. The tech stack in which we will develop the software will be using python as a base programming language (but some components have been built using C++); In this paper we will be making extensive use of a custom branch of the Unreal Game Engine in which we have tailored fitted solutions to our machine learning task. We make use of the Carla, one of the most popular open-source simulators for autonomous driving research;

2.2 Python

Python is a general-purpose language that is commonly compared to a scripting language. It is currently one of the most popular programming languages. The popularity comes from the easiness of writing code and the versatility of the language. Python does not make use of complex operators nor presumes the understanding of low memory operations. Scripts can be developed without advanced programming knowledge. The appeal of the language attracted a lot of attention and in time, developer accommodated to it, building more and more software reliant on this tech.; The language was adopted by both novices and seasoned developers, and its maturity was proven by a plethora of companies as Uber[10], Netflix[11], Spotify, Facebook[12] etc.. Given the newly developed libraries based on python started to attract the attention of machine learning / artificial intelligence enthusiasts. Libraries such as TensorFlow and PytTorch are currently at the top of the AI techs used in production.

2.3 Unreal Engine

Unreal Engine is considered one of the most advanced video game engines in the software development industry. Unreal Engine has unmatched performance in various domains such as gaming, architecture, automotive, broadcast, films and simulations.[13] The engine integrates state-of-the-art techniques to achieve performance and verisimilitude in regards to physics and graphics. The game engine also supports multi-platform support (allowing developers to create a single application which is automatically ported to multiple platforms such as Consoles / Windows / Android / iOS etc.) making it very appreciated among video game developers; The broad toolsets that are available out of the box in Unreal, combined with the easiness of making use of the blueprints system (a proprietary visual scripting utility) skyrocketed the engine among software engineers. Unreal Engine is currently in use by 7.5 million developers, making it one of the most popular game engines [4].

One of the key advantages of the engine is that the engine's source code is completely accessible, allowing users to customize the system in any way they see fit. The current project includes such modifications, our forked branch contains additional machine-learning tools developed to enhance and simplify the generation and capture of data. The subsystems developed were written in the language C++;

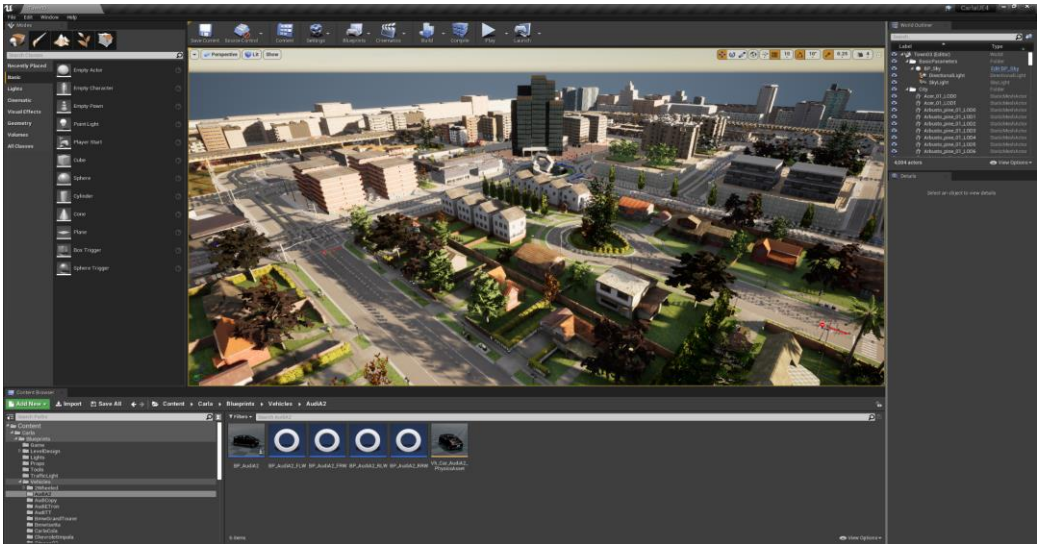

Figure 2. Unreal Engine Scene View

Unreal engine provides a plethora of tools for developers. The simplicity of using the engine makes it a great choice for software reliant on this platform. Upon loading, Unreal Engine creates a default world (in the context of game engines this world is often names as scene). Users can modify these scenes by adding, removing or editing objects in them.

The previously attached image showcases the main tools used in the development of the software. The modes window which facilitates the creation of scene objects. In Unreal Engine such objects are called actors. Actors are objects that can represent a lot of such as cameras, static meshes. They support object translation, rotation and scaling and can be either statically or dynamically created at the start of the game. In our simulator the pedestrians which we will generate will be noted as actors.

Actors can be inspected with the use of the World Outliner and the Details tool. The World Outliner provides the complete list of actors that exist at the moment in the scene (both active and inactive actors). The Details panel offers information regarding various properties of Actors, such as transforms, meshes, materials, default script values, physics, collision body, lightning, attached scripts etc. Unreal Engine also provides users to inspect these properties at run time by allowing developers to run a dedicated Simulate mode in which the players are instantiated but at the same tame the developer has the option to either inspect objects in the game world view or play the scene as intended. Assets of the Unreal Engine project can be found making use of the Content Browser, allow users to scan through the imported or created project files. Assets are serializable objects which can be stored in the file system and can be inserted into scenes.

Unreal Engine permits the use of the box navigation tools which can be attached to any actor existent in a scene. The native navigation system available is based on navigation meshes. These meshes are areas that mark accessible grounds for our path finding algorithms. Navmesh maps can either be baked before the project is built or can be dynamically created but at the cost of performance. Unreal Engine currently makes use of A* path finding algorithm to identify an optimal and correct path to a given destination. The engine employees two different techniques to avoid collisions between agents in large crowds. The two methods are Reciprocal Velocity Obstacles (RVO) and Detour Crowds, both techniques come with limitations of their own. [14]. The RVO algorithm is real-time multi-agent navigation technique focused on improving interaction between local agents. In this approach the authors propose the include in the velocity vector of an

agent, a vector component created as a response to the velocities of other agents in the near vicinity. This new method describes a safe and oscillation free movement. As the technique is force based it is possible for agents to become out of bounds as agents could integrate incredibly sharp responses in their movement as to avoid any possible collision.[15] On the other hand, Detour Crowd is a technique which only takes into consideration navmeshes. This guarantees that at every moment agents will be found only on valid navmeshes. But performance is greatly affected by large crowds.

Given the simplicity of the software, large support and popularity of the engine I have decided to make use of Unreal Engine as my main data generation tools and also as a simulator for the predicted behaviors.

2.4 Carla Simulator

Carla is an open-source simulator which was developed to enhance and simplify the research of autonomous vehicles. The library is open source and provides free usable assets (such as pedestrian blueprints, vehicle blueprints, textures, scene objects etc.). The main use of the library was to provide an easy to set up data collection pipeline and a controllable environment in which artificial intelligence could be tested. Carla provides multiple data collecting features such as the ability to simulate multiple cameras (RGB / DVS / Depth) and sensors (LIDAR / GNSS / RSS) but also allows users to create their own world by offering control of environmental factors such as rain and sunlight, allows the generation and manual control of spawned actors (both pedestrians and vehicles) and allows users to integrate automatic movement into actors and providing map generation tool.

Figure 3. Show case of Carla's asset city scenes

Carla is a scalable simulator developed as a multi-client service. Carla can run in both synchronous mode (in synchronous mode server waits for a client tick before updating the simulation step) and asynchronous mode (in which the server is independent of the clients. This method improves performance and allows the connection of multiple clients at the same time but at the expense that clients will always be behind the master node). Synchronous mode is especially useful if the client applications are very slow and they will start missing ticks received from the server. Clients can interact with the world Carla API (a Python API); One of the most important API components, which allows the scene to realistically imitate real life is Carla's traffic manager. The traffic manager is a complex system which has control over the agents that are spawned in the world and over various traffic obstacles such as pedestrian crossing lights, traffic lights etc.

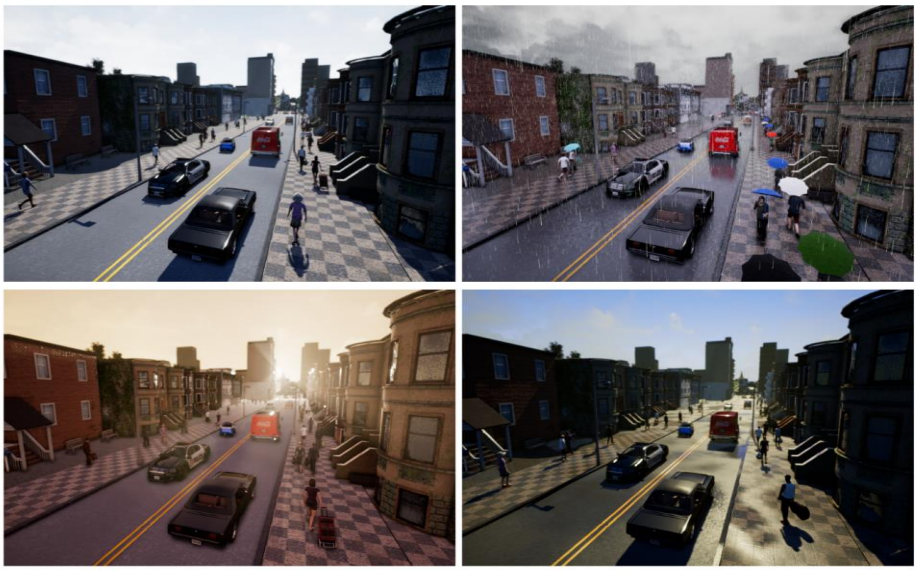

Figure 4. Carla multiple environment settings

Carla features multiple out of the box scripts which can be used by researchers to enhance their developing process. Some of the features that are available are multiple data collection scripts which are based on the aforementioned sensors and a manual steering script which allows users to take control of a vehicle of their choice. One of the most important features is the automatic actor spawn script which can be easily deployed. The spawn script can be configurated to create a significant number of agents (both vehicles and pedestrians). The agents are then assigned to a traffic manager which analyses the positions and the environment and controls the agents accordingly. The Traffic Manager is composed of an Agent Lifecycle & State Management (ALSM) module, a vehicle registry which contains a list of ids of both agents that have an autopilot script attached and inactive agents. And a simulation state which stores the actual information of the actors. The traffic manager acts in three stages. The first stage is represented by the storage and update of the current state of the simulation. In the second stage, the movement of every autopiloted agent is computed. The third and final stage implements the computed velocities of each agent to advance the simulation as intended. The complexity of the system reduces the overall performance of the simulator, the simulator's FPS being reduced from 60 FPS to 30FPS after spawning 30 actors. Pedestrians, the target of our research are controlled by making use of a component entitled AWalkerAIController which is based on Unreal Engine's built-in navigation system.

Figure 5. Image Showcasing Carla Native Pedestrian Spawning

Given the out of the box control which Carla offers users I have decided to make use of to generate and collect data pertinent to creating an AI that is able to control various pedestrians in scenes.

3. Datasets

The datasets used in the creation of this project were collected from a plethora of sources. The three main resources of data used in the research of the trajectory prediction models are ETH Dataset, UCY Dataset, Stanford Drone Dataset and a synthetic dataset generated through the usage of Carla Simulator.

3.1 ETH Dataset

The ETH Dataset was collected as a part of a solution proposed for the problem of pedestrian detection alongside the secondary task ground-plane estimation.[16] The developed system involved a novel technique focused on geometry using Belief Propagation. The dataset which they have built for this task involved video sequences of crowded shopping streets in various weather conditions, the total number of recorded frames being approximately 2300. The data was

recorded using a resolution of 640x480 pixels and a frame rate of 15 FSP making use of a stereo pair of cameras. The dataset was complex as it included various artefacts such as motion blur, lighting bloom and missing contrast.

3.2 UCY Dataset

Motivated by that fact that most crowd modelling techniques worked on the assumptions that crowds can be defined by custom made rules (such as the boids crowd algorithm), rules which are too restrictive to define natural behavior, the authors of the dataset proposed an example-based crowd simulation technique. The authors decided to develop a technique to create a database containing simulations of crowds and their trajectories. The researchers tried to mathematically model the behavior of humans in crowds by developing influence functions, functions which would describe the reaction of humans. Developing such a system would require the creation of an example database. As such they recorded videos of several crowds shot from the rooftops of buildings. The videos have an approximate duration of 5 minutes, every frame being manually ad noted. As real-life crowds are very diverse a lot of different behaviors could be investigated proving to be a very useful dataset for understanding human crowd movement. .[17

3.3 Stanford Drone Dataset

The authors of the dataset wanted to create a performant trajectory tracking system which could be applied on multiple entities. The reasoning behind compiling on the largest datasets for trajectory predictions is that in order to create advanced trajectory forecasting methods we would firstly need to concentrate on the creation of quality data before reasoning about how to mathematically model human behavior. They introduce multiple notions such as “social sensitivity” and “navigation styles” which can be use in order to improve previous trajectory prediction systems.[18]

The Stanford Campus dataset consists of the first large-scale dataset developed for trajectory prediction. The dataset contains both videos and images of multiple classes of entities that are either standing still, moving or interacting with other targets and the environment. The dataset consists of more than 19000 targets. The authors wanted to record multiple types of interactions and they categorize them as either target-target interactions (which would refer to behavior of humans / vehicles avoiding collisions) and target-space interactions (reactions of targets to the environment. such as round-a-bout behavior of vehicles, how pedestrians avoid walls etc.). The dataset was recorded with a flying drone that hoovered over Stanford's University Campus. The authors focused on collecting data at peak times in order to capture more significant crowd interactions. The videos have been filmed from an altitude of 80 meters and the resolution of the images is 1400x1904.

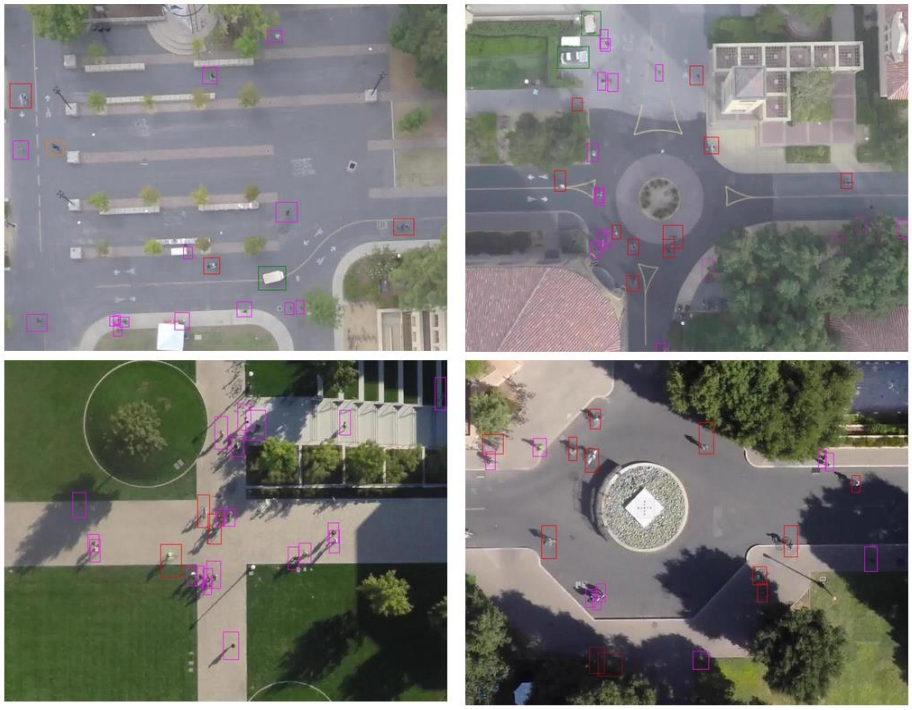

Figure 6. Stanford Drone Dataset Examples

3.4 Synthetic dataset

By making use of the Carla simulator, I was able to generate high quality data by creating episodes of pedestrian crowds in a controlled environment. The actors were natively controlled by Unreal's pedestrian navmesh algorithm (reduced performance but always correct result) which will be used to train a deep neural network model. As such, we will work on the assumption that the Detour Crowd algorithm is similar enough to human behavior.

4. Related work

4.1 Social LSTM

Humans tend to adapt their movement based on their interactions with other humans and objects in a scene; The difficulty of predicting the trajectory come from the innate behavior of humans, their instinctive actions which are sometimes seen as unpredictable; As such the final movement is influenced by a plethora of factors which have to be taken into account as to make a correct prediction; Previous work in this domain showed success when models took into account inferred knowledge regarding the environment (such as location of sidewalks, location of stairs, areas covered by grass) yet the approaches where limited by two assumptions: the usage of custom made function which replicate very specific interactions rather than learning them through a data-first technique and focusing on interactions between close subjects rather than taking into account that humans also tend to make decisions in movement to not only avoid local collisions but future as well;

Motivated by the malleable nature of the human trajectory the authors proposed a LSTM based model which jointly predicts al the trajectories of humans in a certain scene. The model is referred in specialty literature as a Social LSTM model; Our motion is often adapted based on the behavior of people in our vicinity; When facing a crowd, we ought to take into account the overall collision-avoidance behavior between an individual and the crowd; Isolating individuals and analyzing a restricted space certainly not reflect reality.[19]

The datasets used in development of the models are ETH (which consists of two scenes of 750 different pedestrians) and UCY (consisting in two scenes with 786 pedestrians); The datasets are real-world camera recordings of various dense crowds filmed in different environments; The dataset provides real-life crowd behavior and specific social interactions such as dispersing and forming crowds, couples that are walking together and groups that cross each other;

In their approach, the authors make use of an assumption that the data has been prep-processed to remove all visual data and keep only the geometrical information (coordinates at each

step in time); They note the (x_i, y_i) to be the coordinates of a certain person at the time instance t_i. The task can the be modeled as following: given the coordinates (x_i, y_i) at t_i, we need to predict the coordinates (x_{i+1}, y_{i+1}) at t_{i+1}. The task can be seen as a sequential task, tasks in which LSTMs perform particularly well; The authors state that a naïve usage of a single LSTM model per person cannot captures sufficient information to accurately predict human behavior;

To solve this issue, a model which takes into account the jointed hidden states of nearby LSTM models; But this approach introduces a new difficulty: adding together the hidden states of individuals in dense crowds and processing them will require an incredible amount of computing power; The authors propose a new representation which pools together data from neighboring states; The technique, suggestively named social pooling, consists in pooling hidden-state information from near people while preserving spatial information by making use of a grid-like structure to combine the data; The scene is sectioned into a number of cells to improve the model's understanding of locality;

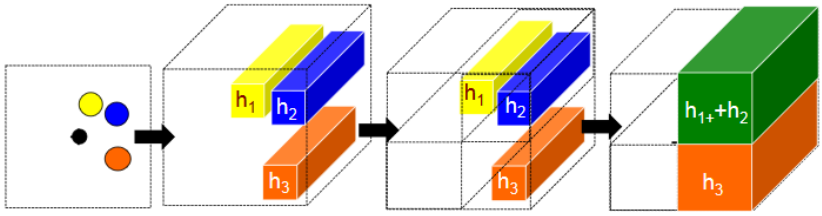

Figure 7. Hidden state concatenation in Social LSTM

In the previous figure we have the representation of the social polling operation; The individuals whom hidden states are h1, h2 are localized in the same cell and as such their hidden states are concatenated;

The hidden state h_i^t consists in the representation of the i^{th} person at the moment of time t. With the hidden-state dimension D_{im} $N_{neighbors}$ (the neighborhood size), the trajectory tensor H_t^i is computed as following:

$$H_i^t(m,n,:) = \sum_{j \in N_{neighbors}} 1_{mn}\left[x_t^j - x_t^i, y_t^j - y_t i\right] h_{t-1}^j, \text{ where}$$

$$1_{mn}[x,y] \; indicates \; if \; the \; coordinates \; (x,y) \; are \; in \; the \; (m,n) \; cell \; of \; the \; grip - map$$

Afterwards, the pooled social hidden state tensor and the coordinates are embedded into two different vectors, namely a_i^t , e_i^t.The embeddings are then concatenated and fed into a LSTM cell; It is important to note that in the case of social LSTM, the hidden states of the neighboring humans are coupled and as such the back-propagation step is jointly applied;

The resulting model offered what at the time was a state of art performance; Combining the information multiple individuals helped the model to get a better grasp on the invisible complex interactions between humans in crowds; It has set a baseline for multiple data-driven approaches (mostly based on LSTMs) to come; Inspired by the novelty introduced by Social LSTMs future techniques to come included more and more informational context into model as to improve their overall knowledge of the environment;

4.2 Social GAN

The human brain analyses a plethora of options before deciding which actions to take. This behavior is innate and subtle, and it is a proof of our evolution as a species. The authors of the research define three properties of human behavior as being vital to the verosimility of predictions: predictions should be interpersonal (meaning that moving entities that are near each other should make decisions based on each other's actions), socially acceptable (the authors state that some trajectories are physically possible but humans do not consider them as acceptable. In real life we can observe that individuals in a crowd tend to not cross a certain threshold of proximity of one another) and multimodal (meaning that there is no absolute path to choose but a plethora of choices to choose from). As such the authors propose a method to construct not only one possible outcome for human trajectory predictions but a variety of futures. The model introduces a novelty architecture in trajectory prediction as the authors combined two different deep learning techniques: recurrent neural networks and generative adversarial methods. [20]

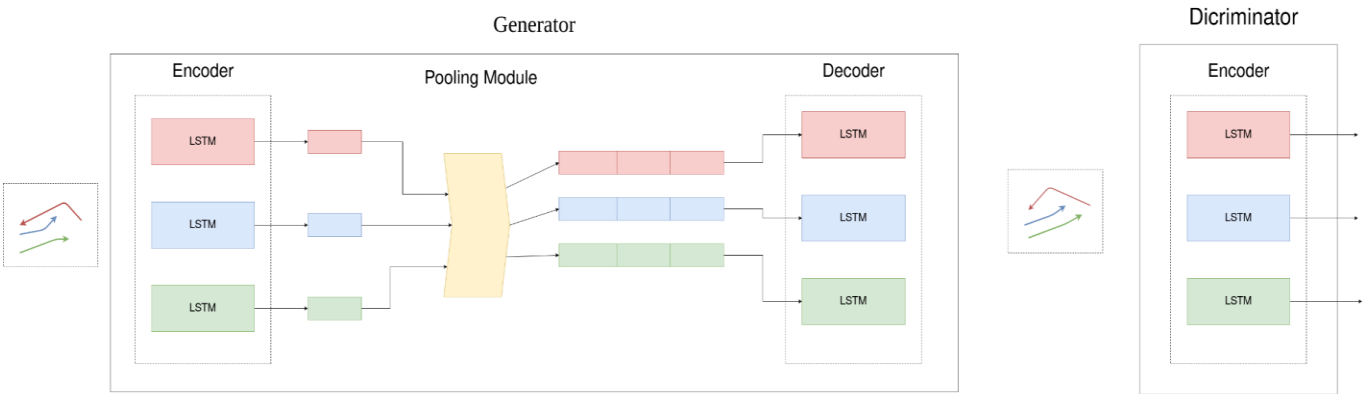

Figure 8. Social GAN Architecture

The problem that the authors wanted to resolve can be interpreted as

Given $X = X_1, X_2, \ldots, X_n$, where $X_i = ({x_i}^t, {y_i}^t), t = t1, \ldots t_{obs}$, we need to predict

$Y = Y_1, Y_2, \ldots, Y_n$, where $Y_i = ({x_i}^t, {y_i}^t), t = t_{obs+1, \ldots} t_{pred}$.

In other word, given the coordinates of the agents at various points in time, the neural network should output the future coordinates of all the agents involved in the scene.

The model architecture is composed of two major components: a generator and a discriminator. The generator is based on three different subcomponents which include an encoder, a pooling module and a decoder.

As a first step, the locations are forwarded through a single layer perceptron to get a spatial embedding of the agent positions. The coordinates are then pushed through an LSTM-based encoder. To improve upon the technique Social LSTM proposed (which added the hidden states of actors found in the same grid square) the authors propose a pooling module in which the relative coordinates of each agent to the target agent are introduced into a multi-layer perceptron followed by a MaxPool operation. The results are then concatenated with the previous hidden states offered by the encoder, and all the data processed by a decoder which transforms the concatenated results into possible trajectories.

The second component, the discriminator is actually a sequence encoder which transforms the input trajectories and classifies them as real trajectory or phony one. A multi-layer perceptron is applied on the encoder final hidden state to obtain the classification score.

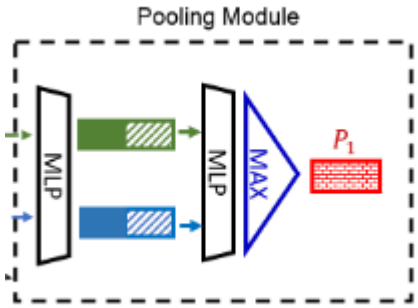

Figure 9. Pooling module employed in Social GAN

4.3 Future Activity Prediction

Motivated by the task of predicting the trajectories of humans in videos the authors proposed a novel technique to analyze the video – information; The authors proposed a novel technique which understands rich visual information and outputs the actions which people indulge into; To facilitate the training of the model they authors made use of an additional task which consists of predicting the location where a certain activity will take place.[21]

The multi-task learning model is composed of modules which can tackle both learning future trajectories and future person-environment interactions at the same time; The authors made use of a rich semantic representation of pedestrians which contains information about the visual appearance of the agents, their movements and their interactions with the environment; For the auxiliary task the environment is mapped onto a grid structure and the model has to predict the patch in which the activities will take place; The authors made use of the assumption that the model can start from an initial state in which the bounding boxes of pedestrians and various objects in the scene are already known; The model observes the positions of the objects between $[1\,;t_{obs}]$ and predicts their position and possible actions;

The predictor is composed of the following modules: a module which has the task of extracting information from the person's behavior (named behavior module), a module which analyses the interactions between people and their surroundings (named interaction analysis module), a trajectory generator and an activity predictor;

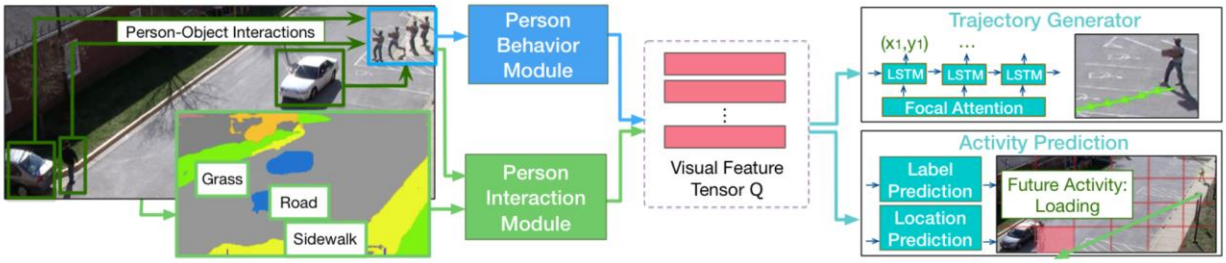

Figure 10. Showcase of model's architecture

The behavior module encodes visual data of the people in videos; The authors purpose that rather than reducing an entity to a single point, keeping data regarding appearance and behavior would improve overall model knowledge; To identify the changes of person's appearance a pre-trained object detection model; Furtherly, for each individual in the scene the feature is averaged along the spatial dimensions and forwarded towards an LSTM encoder; For detecting and analyzing body movement another pre-trained model was been used, model which was trained on the MSCOCO dataset; The extracted feature is then forwarded into a linear transformation and the result is fed into an LSTM encoder which breaks the input into features of shape $T_{obs}\ x\ d$, where d is the hidden size of the LSTM;

The interaction module analyses the video stream, taking into account interactions between people and the scene, between people and objects and also people and other people; Each of the three types of interactions are differently processed inside the module; Person-scene interactions are processed with a pre-trained model focused on image-segmentation which was trained on 10 distinct classes (grass, roads, sidewalks, road signs, and many other common street objects). Each frame is transformed into a binary mask and are averaged across the temporal dimension; Two convolutional layers are applied on the previous result to obtain scene CNN features of two different scales; A pooling on the convolutional feature maps on each person's xy-coordinates; The information is fed into an LSTM encoder for capturing temporal data and to obtain actor-scene interaction features. In order to analyze person-object interaction a custom geometrical relation between a person's bounding box and the objects is used;

$$G_k = [\frac{|x_b - x_k|}{Wb}), log\ log\ \left(\frac{|y_b - y_k|}{h_b}\right), log\ log\ \left(\frac{w_k}{w_b}\right), log\ (\frac{h_k}{h_b})], \qquad where$$

$$x_b = x\ coordinate\ of\ a\ person$$

$$y_b = y\ coordinate\ of\ a\ person$$

$$w_k = width\ of\ the\ person's\ bounding\ box$$

$$h_k = height\ of\ the\ person's\ bounding\ box$$

$$x_K = x\ coordinate\ of\ the\ k-th\ object\ in\ the\ scene$$

$$y_k = y\ coordinate\ of\ the\ k-th\ object\ in\ the\ scene$$

$$w_b = width\ of\ the\ object's\ bounding\ box$$

$$h_b = height\ of\ the\ object'sbounding\ box$$

A one-hot encoding is used to get a feature in the space of $R^{K\ x\ N_o}$, in case of object type detection.

The trajectory module takes into account both the previous trajectory of a person and a trajectory embedding obtained by the use of the following formula:

$$e_{t-1} = tanh\ tanh\ \{W_e[x_{t-1}, y_{t-1}]\} + b_e \in R^d, where$$

$$[x_{t-1}, y_{t-1}] = the\ trajectory\ prediction\ of\ time\ t-1$$

$$W_e, b_e = learnable\ parameters$$

The previous embedding is the fed into another LSTM embedder to obtain a trajectory prediction; Afterwards by stacking together the hidden states of all the encoders, named $Q \in R^{MxT_{obs}xd}, where\ M = 5$ is the number features and d is the size of the hidden layers of the LSTM; A LSTM decoder is used on the previous tensor to predict the trajectory of the person. The initialization procedure used by the authors was making use of the last state of a person LSTM encoder; This hidden state is then passed to a fully connected layer;

$$h_t = LSTM(h_{t-1}, [e_{t-1}, q_t])$$

A custom effective focal attention technique is used to this end. A correlation matrix $S^T \in R^{MxT_{obs}}$ is computed at each time step; For each entry we apply:

$$S_{ij}^t = h_{t-1}^T \,.\, Q_{ij:}, \ .$$

Afterwards, two focal attention matrices are used:

$$A^t = softmax(S_i^t) \ \in \ R^M \ ;$$

$$B^t = [softmax(S_{1:}^t), \dots, softmax(S_{M:}^t)] \ \in \ R^{MxT_{obs}} ;$$

Resulting the feature vector: $q_t = \sum_{j=1}^{M} A_j^t \sum_{k=1}^{T_{obs}} B_{jk}^t \, Q_{jk:} \in R^d$;

The authors argue that since the trajectory predictor only outputs one prediction, errors may cumulate over time; They propose introducing an auxiliary task (predicting future activities) alongside a third task which is the prediction of the location where the activity is going to be held; The problem is split into two different subtasks with a similar goal; One of the is a classification which will output the predicted Manhattan Grid in which the activity will take place and the other one is a regression task which will predict the deviation of the person from the center of the previously predicted grid block to the final coordinate of the person; The additional regression task provides a more precise prediction for the localization of the individual and they argue that will offer the model more information to work with rather than just a patch of scene; The previously described scene CNN features are concatenated with the final hidden state of the LSTMs and are forwarded into two different convolutional layers (one which performs the classification task and another one which performs the regression);

The activity label prediction is a multi-class classification problem; Individual could be multiple activities at the same time (for example walking and carrying activities); The future activity probabilities are computed by making use of the concatenated final hidden states of the encoders:

$$cls_{act} = softmax(W_a . [Q_{1Tobs:}, \dots, Q_{MT_{obs}:}])$$

The model was trained and tested on ActEV/VIRAT. The dataset includes 455 video recordings taken at 30 fps, recordings which have a resolution of 1920x1080. Each of the videos include more than 12h of footage. When compared to similar models developed for the task of future activity detection, it current model greatly outperforms its peers. Even when adding noise input as to mimic a real life environment the model manages to output satisfying results given the complexity of the issue;

	Method	ADE	FDE	move_ADE	move_FDE
Single Model	Linear	32.19	60.92	42.82	80.18
	LSTM	23.98	44.97	30.55	56.25
	Social LSTM	23.10	44.27	28.59	53.75
	SGAN-PV	30.51	60.90	37.65	73.01
	SGAN-V	30.48	62.17	35.41	68.77
	Ours	**17.99**	**37.24**	**20.34**	**42.54**
	Ours-Noisy	34.32	57.04	40.33	66.73
20 Outputs	SGAN-PV-20	23.11	41.81	29.80	53.04
	SGAN-V-20	21.16	38.05	26.97	47.57
	Ours-20	**16.00**	**32.99**	**17.97**	**37.28**

Fig 11. Comparisons between various trajectory prediction techniques

The authors conclude that combining a comprehensive representation of visual features of people and embedding an auxiliary task into prediction model resulted in a state-of-the-art technique for predicting future trajectories and human activities.

4.4 Future Object Location

The prediction of future events in videos is one of the most studied computer vision problems in present; The general lack of data for training pertinent predictors motivated the authors of the paper to publicly release a new dataset, namely Citywalks, which comes hand in hand with a new expression of the object trajectory prediction problem and a custom model which fusions both visual and temporal features to obtain accurate predictions on the previously described problem; [22]

The problem formulation which they propose is that of multiple object forecasting which compared to the standard task of multiple object tracking (MOT) rather than outputting the bounding boxes of objects in the current frame, the task is to predict the future bounding boxes of objects; The authors reason that in MOT an acceptable performance can be obtain through the use of high-quality object detectors and simple linear velocity motion assumptions; However this approach does not perform well in case of complex movement patterns and making use of more

sophisticated means usually introduces overfitting as the datasets available at the time are of small dimensions and as a counter measure propose the creation of a new and larger dataset;

Pedestrian trajectory forecasting has been studied in numerous occasions, the authors of the paper referencing the pioneering social LSTM model which takes into account complex interactions between nearby objects; They also state that current models oversimplify the visual information that we have available and praise the novelty of introducing an auxiliary task and encoding complex visual cues

The authors formulate the problem as following: given a sequence of n video frames marked as $f_1, f_2 \dots f_n$; Given the t^{th} frame f_t, each object has a correspondent set of coordinates $b_t^i = (x_t, y_t, w_t, h_t)$ which represents the centroid and the height and width of the bounding box. Given all the frame-level detections the task is to associate each detected object with a unique label; This task is extended by the authors to include future bounding box prediction as following: Given $f_{t-p}, f_{t-p+1}, \dots, f_t$ with associated object detections $b_{t-p}^i, b_{t-p+1}^i, \dots b_t^i$, the problem is to predict the future bounding boxes $b_{t+1}^i, b_{t+1}^i \dots b_n^i$;

The dataset is composed by 358 different videos containing 21 different European cities in various weather conditions: rain, snow, fog, sun etc. Each video has a length between 50 and 100 minutes and the scenes are both indoors and outdoors; The pedestrians are detected by making use of two pre-trained models (a YOLOv3 and Mask-RCNN) on the MS-COCO dataset which seemed to generalize well on the obtained dataset. The frames are upscaled / downscaled according to the input sizes required by the models;

To solve this difficult challenge the authors proposed a model entitled STED, which is a encoder-decoder model which takes into account both visual and temporal features; The model is composed of three deferent modules: A bounding box module which is based on a gated recursive unit that extracts features from the bounding boxes in the past, a convolutional neural networks module which extracts movement related features from the continuous frames and a decoder module.

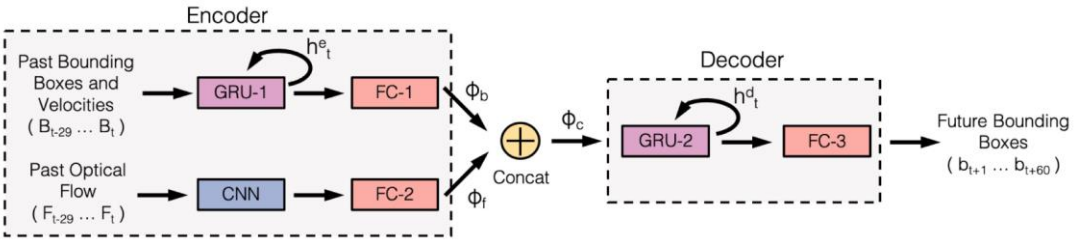

Figure 12. Scheme of the sequence to sequence model

The authors introduced a new dataset for trajectory prediction to facilitate future research and have shown that models trained on the newly collected data perform well on popular trajectory tracking benchmarks such as MOT-17; Furthermore, the proposed model managed to predict the bounding boxes of objects up to two seconds, understanding nonlinear trajectories and complex movement; This work shows promise in building more advanced techniques to enhance object-tracking by helping them avoid common mistakes such as occlusions or missed detections;

4.5 Pedestrian Movement Understanding

Given the high difficulty of data-driven model to understand the hidden and complex interactions between pedestrians, the authors of the paper propose a new approach for pedestrian trajectory forecasting, an approach which nowadays is considered one of the baselines solutions for this task; The new predictions model, named Behavior-CNN, takes into account multiple factors such as the interactions between moving and stationary humans and also historical information; [23]

The novelty of the approach starts for the input data; Rather than feeding direct frames into encoders, the walking path of the pedestrians are transformed into displacement vectors and afterwards all the different trajectories are combined into a displacement volume; The input space is transformed into a grid-map to improve ease the difficulty of understanding locality; Let X,Y be the dimensions of the grip-map; Let $p_1, p_2 \dots p_n$ be the pedestrians in a scene, $t_1, t_2 \dots t_M$ be uniformly sampled moments to be used as inputs, where t_M is the current time point; The normalized location of a pedestrian p_i is noted as $1_i^m = [x_i^m, y_i^m]$ where x_i^m, y_i^m are the coordinates of the i^{th} pedestrian at the moment of time m; Afterwards, the positions of the

pedestrians are encoded into $2M$ dimensional displacement vector $d_i = [1_i^M - 1_i^1, 1_i^M - 1_i^2, .. 1_i^M]^T \in R^{2M}$;

To prepare the input for the convolutional neural network a 3 dimensional vector is constructed from d_i, noted as $D_f(x_i^M, y_i^M) = d_i + 1^T$, where 1^T represents an all one vector; The all one vector is added because previously the range values of the displacement vector were between $(-1; 1)$.By adding a unit stationary pedestrians will now be differentiated from the background;

The CNN is then fed with the displacement vector and tries to predict a future displacement vector; The CNN is composed of three convolutional layers followed by a max-pool operation with a large receptive field to allow the network to understand relationships between distant points; Afterwards the result is combined with a location bias which is learnable; The inputs continue to be fed into three convolutional layers and afterwards a deconvolution is used to obtain the final prediction displacement volume.

The authors argue that due to the high sparsity of the inputs, when the neural layers are instantiated randomly and trained together, the network tends to converge to a local minimum; As such, they have made use of a layer-by-layer training; The loss function used to train this model was is L2 loss, the averaged squared distance between the predicted displacement volume and the ground truth;

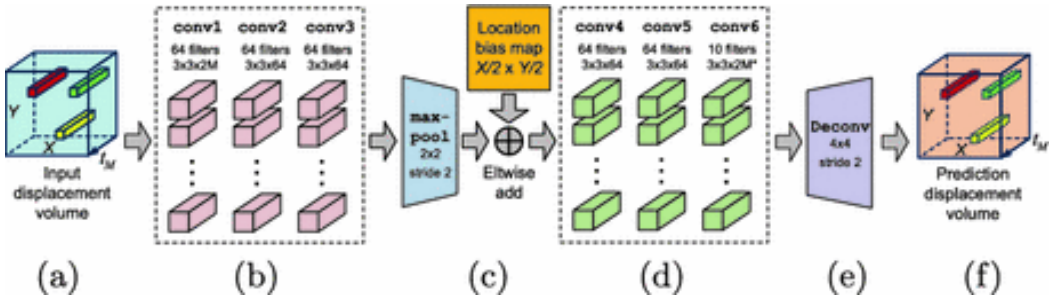

Figure 13. Architecture of the neural network model

The previous model resulted in a state-of-the-art model compared to the previously existing trajectory predictions; The encoding scheme used helped the neural network understand pedestrian behavior;

4.6 Crowd Interaction and Trajectory Prediction

The behavior of pedestrians is encoded in the complex behavior of humans, behavior which is strongly influenced by the environment; The authors propose a novel technique to analyze the interactions between humans as help models understand what drives our moving patterns, by making use of a crowd interaction deep neural network (CIDNN). This neural network takes into account de displacement of agents found in the same proximity. The authors criticize the inability of Behavioral CNNS to model future interactions between agents and praise the Social-LSTM approach which takes into account the relationships between closely found agents; Even so, social pooling lacks the ability to understand information which we subconsciously process such as the spatial and motion information of people in close proximity; [24]

In their paper, the authors propose a model which sequentially predicts the displacement of pedestrians between two sequential frames; The assumption on which the model acts is that the movement of pedestrians is dependent on its motion data (such as speed, orientation and acceleration), the motion data of other pedestrians and the spatial affinity between pedestrians; The architecture is composed of four different modules : a location encoder module, a motion encoding module, crowd interaction module and a displacement prediction module;

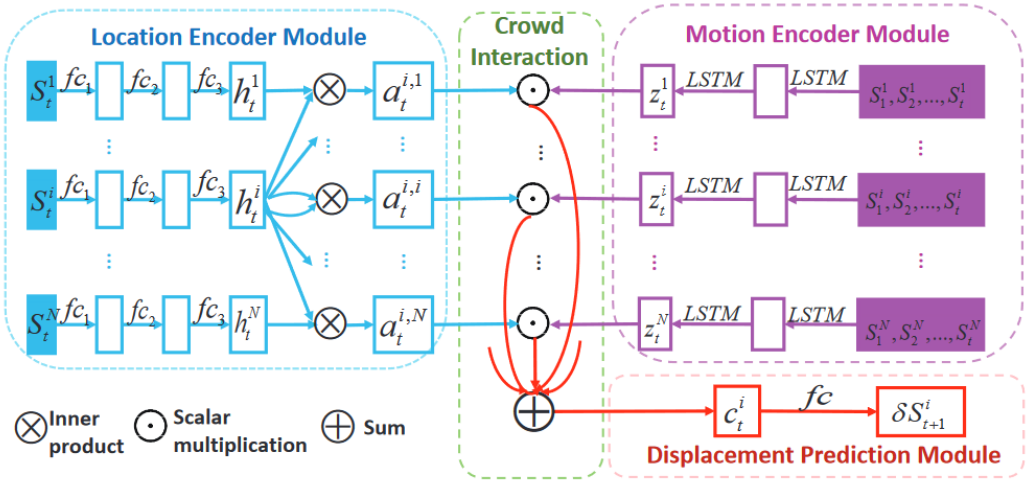

Figure 14. Showcase of the proposed model

The motion encoder module is developed to offer a deeper understanding of the movement patterns of pedestrians; It takes into account travel history, orientation, the velocity and acceleration of each individual; Given the sequential nature of the data a LSTM network was used to encode the information. Two stacks of LSTM cells are used to output the motion encoding; The location encoder module in developed to compute the spatial affinity between pedestrians. A simple gaussian kernel could take into account only distances which is not at all reflecting real life behavior; A more complex and learnable kernel is the proposed: a multi-layer perception composed of layers which is feed with the pedestrian coordinates at different timestamps; The crowd interaction module is a simple operation between the outputs of the previous two layers combined with a scalar multiplication; This combination will ensure that both the spatial affinity and movement information are picked up by the predictor; The displacement prediction module is composed of multiple LSTM cells which

The technique was compared with other trajectory prediction models; The datasets used in the comparisons were the GC dataset, the ETH and UCY datasets, the CUHK crowd dataset (which contains videos of crowds with various speeds and densities in multiple environments) and a subway station dataset (which consists of a 30 minute sequence of video footage taken inside the NY Grand Central Station, containing more than 40000 key points trajectories); The metric used to compare the models was the average displacement error;

dataset	const acc	SF [23]	S-LSTM [1]	B-CNN [25]	SRGP [7]	Ours
ETH	0.80	0.41	0.50	0.35	NA	0.09
HOTEL	0.39	0.25	0.11	0.18	NA	0.11
ZARA 1	0.47	0.40	0.22	0.20	NA	0.15
ZARA 2	0.45	0.40	0.25	0.23	NA	0.10
UCY	0.57	0.48	0.27	0.25	NA	0.12
GC	0.099	0.033	0.020	0.024	NA	0.012
CUHK Crowd	0.046	NA	0.0341	NA	0.029	0.008
subway station	0.064	NA	0.0335	NA	0.031	0.016

Figure 15. Comparisons between different techniques and the method proposed

The authors concluded that indeed adding more information than the proximity of

individuals into the compotation of trajectories promoted the neural network model to new heights obtaining a state-of-the-art performance on public pedestrian trajectories benchmarks;

5. Development Process

5.1 Data Collection

The first step in developing an artificial intelligence-based system is the collection and processing of data. After profound research I have decided which publicly available datasets to use and how to increase the quantity of data that I can feed into the predication models which I have employed.

5.1.1 Public Datasets

The aforementioned datasets have been imported from the paper Social-GAN which offered a complied structure which included ETH / UCY / Stanford datasets. The data have been reduced only to the coordinates of the moving entities. This has removed scene information and most of the human-environment interaction. This way the AI is incentivized to focus on the geometrical aspects of the problem. The coordinate has been transformed via a publicly available homography which the authors described in their project. The intention was to transform the coordinates of humans into meters as to uniformize the locations across datasets. Other preprocessing steps have not been applied on the gathered dataset as losing so much information for the original dataset already reduced heavily. Adding multiple preprocessing steps will come at the risk of losing all relevant data. Other preprocessing steps have not been applied on the gathered dataset as losing so much information for the original dataset already reduced heavily. Adding multiple preprocessing steps will come at the risk of losing all relevant data.

5.1.2 Synthetic Dataset Generation

The process of generating synthetic data for the machine learning algorithms is reliant on the Carla python API which allows the creation and control of actors thought the use of the exposed Python API. To generate the dataset an episode-based script was used to generate information based on the Carla map which we want to include (for this project I have used the default Carla Simulator scene: Town01), the number of pedestrians that we wish to spawn, to full length of the episodes. Carla Simulator limits the possible location of pedestrian spawning by offering only certain spawn areas. To assure that large crowds are indeed created in a simulation a method of filtering closer spawn points and generating actors based on the area in which the simulation should be concentrated was employed. The algorithm computes the distance of a randomly selected spawn point by Carla and accepts it only if its distance to the point of interest is small enough. If the spawn point is rejected then another spawn location is requested from the Carla traffic manager. Pedestrians are automatically attached an AI blueprint which includes a navigation and collision avoidance system. After running episode for the number of frames requested the script exports the exact positions of each pedestrian at all points in time as to create a similar structure to the previously imported datasets.

A custom script was developed to generate a full-scale scene segmentation. The full scene was split into multiple rectangles on which image segmentation was applied. This generates a map which actors can use to understand the environment in their nearby vicinity. At various coordinates describing a rectangular grid an image segmentation camera is spawned at an altitude which allows it to properly capture every object in the current grid. The captured images contain segmentations of roads, crosswalks, road markings, vehicles, pedestrians, grass areas, buildings, road signs, statues and train tracks. The generated images have been are included in the custom-tailored model which I have proposed as to allow the deep learning model to properly understand the scene.

5.2 Data preprocessing

As previously mentioned, the public datasets with in training the models have been cleared up completely of visual data, reducing the problem to the prediction of the coordinates of the agents. The coordinates have been transformed using a homography to ease the training across different datasets. The segmented images generated by our synthetic dataset were rescaled to 128x128 while maintaining aspect ratio. This change speeds up the training process and is a workaround VRAM limitation.

5.3 Models employed

One of the intended features of the developed software was to provide an intuitive and fast method to swap between various environments. Allowing this required an overhaul of the Python PI exposed by the Carla simulator. This has allowed me to obtain complete control over settings such as loaded maps, weather, number of agents and many other configurations. Carla does not support a custom behavior for the pedestrian trajectories so a special component had to be built in order to allow the integration of the trajectory prediction models. This component allowed the decoupling of agents and the behaviors implemented. It allows the selection of one of three behaviors: Social LSTM based trajectories, Social GAN based trajectory and Social Gan with Awareness based trajectories (a new custom model which we tailored based on some will be described in the following method). Agents now are provided new trajectories to follow at each frame in order to obtain complete control over the forces impregnated on each agent.

The software's role as a machine learning tool is achieved by coupling of the agent behaviors in the agent algorithm selection. A software component entitled title manager supervises the number of agents created and the type of behavior.

One of the intended features of the developed software is to provide a quick method to integrate a test various neural network models in the Carla simulator.

5.3.1 Social LSTM

Social LSTM was one of the first breakthrough in terms of pedestrian trajectory prediction. It was the first model to include the notion of locality in the more literal sense. The Social LSTM approach divides agents based on their current locations in a grid and selects only the hidden states of agents that are located in the same grid when predicting future trajectories. I have decided to use this model as it marks a cornerstone of progress in this task.

I have made use of a pretrained social LSTM on the collected databases Performance wise Social LSTM is unable to sustain more than 15 agents live, the complexity of computing the grid layout of the agents making this approach unsuitable for real-time applications. Having this in mind I have decided to pursue another trajectory prediction model as to create a system which can offer faster inference speeds. Other problems which were observed were the fact that social LSTM becomes unstable after a few reruns, the algorithm starts to forget the original intention of the agents and reassigns different targets as destination points that it previously intended to. This erratic method of resetting the final destination target creates an behavior which does not seem similar to human behavior. The AI is also unaware of the environment, forcing agents to enter usually unpassable terrain such as grass fields, roadways and railroads. To modify this behavior, I have forced the simulator to retarget the AI to the closest point available on the navmesh to preserve the correctness of the actor positions.

5.3.2 Social-GAN

Social GAN is another keystone technique for pedestrian trajectory prediction which impressed with its ingenuity. The novelty of Social-GAN consists of merging sequence prediction models with generative adversarial networks. The authors combined recurrent neural networks which have been consistently offering great results in tasks which require encoding-decoding operations with generative adversarial models. The preferred recurrent neural network cell used in the paper were long short-term memory cells. LSTMs became popular as they are not vulnerable

to one of the most difficult issues of recurrent neural networks: the vanishing gradient problem. LSTMs introduced a “forget” gate which allow the neural network to reset the error flow if needed.

The other main component of the Social GAN model is the generative adversarial network which consists of two neural network which that are in a continuous contest with one another. The generative model tries to understand the data distribution of the pedestrian trajectories while the discriminator model estimates how probable it is that certain coordinates could have been produced by the distribution function which the data follows.

The model was chosen as to include the novelty introduced by the previous paper (importance of locality) while also keeping a fast selective process. For this neural network I have used both a pretrained model offered by the paper authors and calculated the accuracy on the synthetic dataset and also retrained the model on the synthetic dataset. By making use of this method real-time inference has been achieved for almost 60 different agents at the same time. Issues related to the historical data remained as the algorithm only functions on frames of length 12 and does not seem to preserve a lot of information from earlier stages. As the neural network is not fed an scene data the AI is unable to restrict the movement of pedestrians to only accessible areas.

5.3.3 Social-GAN with environment awareness

This custom made deep neural network model is a fork of the previously described Social GAN. This model was introduced to solve the recurring issue of agents starting to get out of the intended bounds. The filtering of the visual data, which reduced environment information, left the neural network unable to understand the concept of out-of-bounds areas (areas which the actors should not be able access). To improve the variability of the model additional scene data was included in the model. The additional data consisted of a map grid (the map was divided into multiple tiles as to allow the model to concentrate only the specific area in which agents can be found) on which I have applied image segmentation. The segmentation provides useful the context information needed for actors to understand traversable and non-traversable paths. This improvement does not allow the model to fully imitate human behavior as only one agent-

environment interaction has currently been accounted for. Timed traversable paths such as crosswalks are currently unknown to the actors as the image segmentation provided are not live images but rather static ones.

Figure 16. Example of map tile on which image segmentation was applied

The Social GAN model now includes an image encoder which is composed of four convolutional layers (the kernel size of each convolutional layer is 5x5) after which I have applied maxpooling (maxpool operation size of 2x2) to reduce the dimensionality of the data and afterward passed into a rectified linear activation function and three fully connected layers. The image encoder kernels. Similar to the convolutional layers after each fully connected layer a ReLU activation function is applied. The image encoder result is concatenated to the spatial embedding of the actor coordinates before being send to the pooling module.

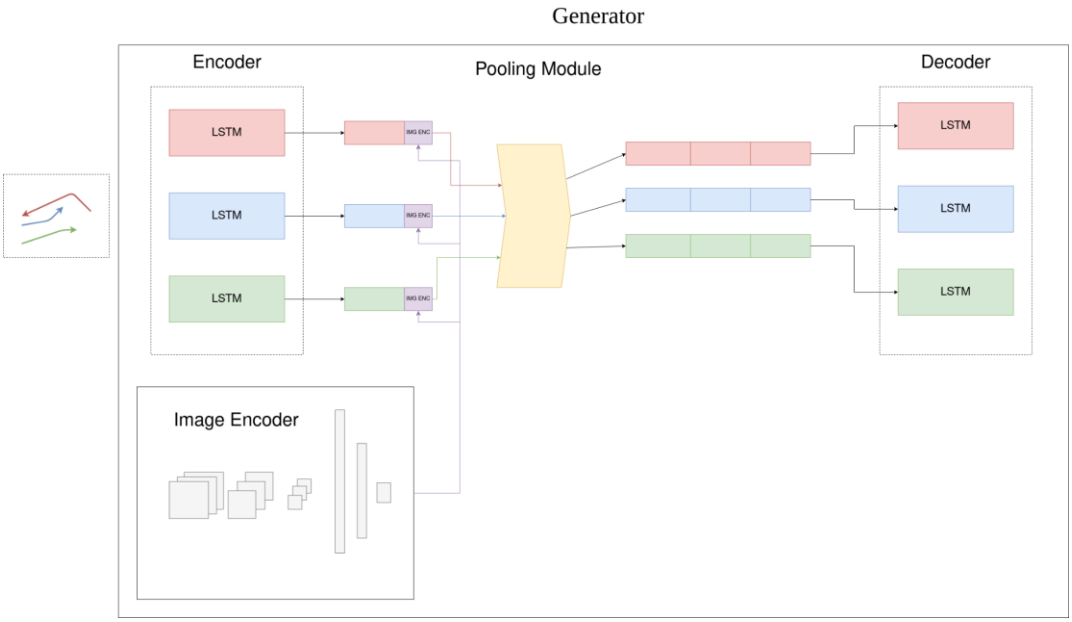

Figure 17. Figure of Social GAN Generator with context awareness

6. Results

The machine learning models were successfully integrated into Carla, pedestrians can be easily spawned by simple delegating one of the three techniques employed. The performance of the system is influenced by the fact that Carla runs in synchronous mode, forcing the server to halt until the client sends a confirmation that it is ready to receive a new frame. As such, communication between server and client limits the performance of the system.

Social LSTM based approach produced the most erratically behavior of the three models, providing both low performance and noisy trajectories. Agents would erratically swap their trajectory based on the last few coordinates provided. The method allowed only for the spawn of 12 agents before reducing to lower than real-time performance. This approach could be suitable for slow renders and other applications that are not reliant on an instantaneous result.

Technique	Nr. Epochs	ADE (train)	ADE (validation)	ADE (test)	FDE (train)	FDE (validation)	FDE (test)	Training Time
Social GAN	60	2.038	4.911	5.301	2.886	8.205	8.945	5h
Social GAN	100	1.607	**4.533**	4.955	2.434	7.617	**7.934**	8h
Social GAN (context awarness)	90	2.475	6.620	7.120	2.800	12.523	13.402	18h
Social GAN (context awarness)	125	2.466	6.617	6.820	2.782	12.519	12.944	24h

Figure 18. Comparisons between the results of the employed techniques

The comparisons between the employed models are related in regards to the average displacement error (the total cumulated error at each step in time between the real trajectory and the predicted one, abbreviated as ADE) and also the final displacement error (the difference between only the final positions of the predicted and real coordinates, abbreviated as FDE).

Social GAN offered both better performance and a more realistic result. Agents were not as prone to changing directions even though the issue is still in effect. Social GAN does not seem to remember actions of agents over larger time ranges and as such can also result in the abnormal behavior felt when using Social LSTM.

To control the erratic behavior and improve the verosimility of the simulation, I have forced the navigation component of the AI to respect navigable paths. As such, agents are now forced to respect unwalkable areas. This offers great result with a minimum implication from the developer. The software easily supported a real-time application with 100 agents. The agents manifested avoidance behavior proving that the model learned the collision-avoidance from the dataset.

Figure 19. Social GAN Controlled Actors not respecting scene areas

Figure 20. Social LSTM actors manifesting collision avoidance behavior

Figure 21. Social GAN Actors restricted with custom scripts to walk on navigable paths

Figure 22. Showcase of 100 agents in real-time using SGAN

Social GAN with awareness did not perform better than the original. Additional data seemed to confuse the neural network model. The model suffered the same vulnerabilities as the previous ones in regards to historical data. The model did not seem to discern yet between navigable and non-navigable paths, defeating the purpose of the context awareness. It is possible that with additional tuning and adding more data variation the model would perhaps understand the challenge better.

7. Conclusions and further improvements

The techniques employed created a sufficiently realist simulation of pedestrian trajectories to include in short clips. The inability of the neural network to maintain the same target trajectory through the course of the simulation can be observed across all the trained models. To solve this issue gracefully a simple script which restricts the possible movements to

I consider that the research was a success having developed a system which can easily generate synthetic datasets as to augment de training of neural network models. The system can be easily coupled with any trajectory prediction technique, having already three different models integrated in the system: Social LSTM Model / Social GAN Model / Social Gan with augmented scene awareness. Comparisons between the models employed show that the augmented network did not perform better than the vanilla technique.

For future work I would propose adding the relative coordinates of an agent to the image segmentation grid selected to improve the model's understanding of agent locality. By providing this additional data the tendency of the AI to propose unsupported trajectories could be removed allowing the model to be fully run without the intervention of custom script logic. Another technique which could be useful is to integrate the image segmentation data after the pooling operation as to reduce the unnecessary data which is pushed into the single layer perceptron. This could improve the overall performance as the visual data is not yet required at this step, allowing the neural network to understand and focus only on the necessary data at that point.

8. Bibliography

[1] Cameron Buckner, Rational Inference: The Lowest Bound

https://onlinelibrary.wiley.com/doi/full/10.1111/phpr.12455 11.11.2020

[2] Abraham Maslow, A theory of human motivation, Martino Fine Books, 2013

[3] Corina E. Tarnita, The ecology and evolution of social behaviors in microbes,

https://jeb.biologists.org/content/jexbio/220/1/18.full.pdf 13.03.2021

[4] H. F. Harlow, R. O. Dodsworth, M. K. Harlow, Total social isolation in monkeys,
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC285801/?page=3 10.01.2020

[5] – CS221: Artificial Intelligence: Principles and Techniques, Stanford University

https://stanford-cs221.github.io/winter2021-extra/modules/general/Introduction-to-the-course.pdf 28.03.2021

[6] – Andrew Pollack, Technology: The computers as translators, New York Times National Edition

https://www.nytimes.com/1983/04/28/business/technology-the-computer-as-translator.html

29.03.2020

[7] – Nils J. Nilsson, The Quest for Artificial Intelligence, Stanford University

https://ai.stanford.edu/~nilsson/QAI/qai.pdf 29.03.2020

[8] – Stuart Russell, Peter Norvig, Artificial Intelligence: A modern approach, Prentice Hall

https://www.cin.ufpe.br/~tfl2/artificial-intelligence-modern-approach.9780131038059.25368.pdf

31.03.2020

[9] – D.O. Hebb, The organization of behavior: A neuropsychological theory, McGill University, Lawrence Erlbaum Associates,2002

[10] - https://eng.uber.com/tech-stack-part-one-foundation/ 22.08.2021

[11] - https://netflixtechblog.com/python-at-netflix-bba45dae649e 22.08.2021

[12] - https://realpython.com/world-class-companies-using-python/ 22.08.2021

[13] - https://www.gamesindustry.biz/articles/2020-01-16-what-is-the-best-game-engine-is-unreal-engine-4-the-right-game-engine-for-you 22.08.2021

[14] - Francesco Sapi, Hands-On Artificial Intelligence with Unreal Engine, Packt

[15] - Jur van den Berg, Ming Lin, Dinesh Manocha, Reciprocal Velocity Obstacles for Real-Time Multi-Agent Navigation

https://ieeexplore.ieee.org/document/4543489 22.08.2021

[16] - Andreas Ess, Bastian Leibe, Luc Van Gool, Depth and Appearance for Mobile Scene Analysis

https://ieeexplore.ieee.org/document/4409092 22.08.2021

[17] - Alon Lerner,Yiorgos Chrysanthou, Dan Lischinsky, Crowds by Example, https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.107.3028&rep=rep1&type=pdf

22.08.2020

[18] - A. Robicquet, A. Sadeghian, A. Alahi, S. Savarese, Learning Social Etiquette: Human Trajectory Prediction In Crowded Scenes

https://cvgl.stanford.edu/projects/uav_data/ 13.01.2021

[19] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, Silvio Savarese, Social LSTM:Human Trajectory Prediction in Crowded Spaces

http://vision.stanford.edu/pdf/alahi2016cvpr.pdf 13.01.2021

[20] Agrim Gupta, Justin Johnson, Li Fei-Fei Silvio Savrese, Alexandre Alah,

Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks, Stanford University, Ecole Polytechnique Federate de Lausanne

https://arxiv.org/pdf/1803.10892.pdf 13.01.2021

[21] Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander Hauptmann, Li Fei-Fei, Peeking into the Future: Predicting Future Person Activities and Locations in Videos

https://arxiv.org/abs/1902.03748v3 13.01.2021

[22] Olly Styles, Tanaya Guha, Victor Sanchez, Multiple Object Forecasting: Predicting Future Object Locations in Diverse Environments

https://arxiv.org/pdf/1909.11944v2.pdf 13.01.2021

[23] Shuan Yi, Hongsheng Li, Xiaogang Wang, Pedestrian Behavior Understanding and Prediction with Deep Neural Networks

https://link.springer.com/chapter/10.1007%2F978-3-319-46448-0_16 – 25.01.2021

[24] Yanyu Xu, Zhixin Piao, Shengua Gao, Encoding Crowd Interaction with Deep Neural Network for Pedestrian Trajectory Prediction

https://openaccess.thecvf.com/content_cvpr_2018/papers/Xu_Encoding_Crowd_Interaction_CVPR_2018_paper.pdf 25.01.2021

9. Figure Table

Image referenced from https://arxiv.org/pdf/1909.11944.pdf

Figure 13 - Figure showcases the neural network model employed in the paper Pedestrian Behavior Understanding and Prediction with Deep Neural Networks

Image referenced from https://link.springer.com/chapter/10.1007/978-3-319-46448-0_16

Figure 14 - Figure showcases the proposed model in the paper

Image referenced from
https://openaccess.thecvf.com/content_cvpr_2018/CameraReady/2136.pdf

Figure 15 - Figure showcases comparisons between the proposed model and previous state-of-the-art approaches

Image referenced from

https://openaccess.thecvf.com/content_cvpr_2018/CameraReady/2136.pdf

Figure 16 - Figure showcases a map tile on which image segmentation was applied

Figure 17 - Figure showcases Social GAN Generator with context awareness

Figure 18 – Figure showcases comparisons between the results of the employed techniques

Figure 19 – Figure showcases Social GAN Controlled Actors not respecting scene areas

Figure 20 – Figure showcases Social LSTM actors manifesting collision avoidance behavior

Figure 21 – Figure showcases Social GAN Actors restricted with custom scripts to walk on navigable paths

Figure 22. Figure showcases 100 agents in real-time in Carla Simulator using SGAN