Hellebaut Jordy

Deepfake (talking painting)

Graduation work 2020-2021

Digital Arts and Entertainment

Howest.be

INHOUD

A. ABSTRACT

I turned an image of the Mona Lisa into a talking painting using neural networks. To do this, a full software service was built that consists of the following steps. First a piece of text provided by the user is turned into speech using a Text to Speech (TTS) system. This TTS will create a wav file that will be used in the next model called lip-sync. Lip-sync will take a source video and the wav file. It will then turn them into a list that indicates the positions of the key areas of the mouth at each point in time. As last you have the deep fake which turns the keyframes in a natural looking video of Mona Lisa. Finally the original wav file is combined with the produced video. Each step was put into practice using already available open source code.

B. INTRODUCTION

A talking painting is a video where we take an existing non moving painting, like for example the Mona Lisa, and turn it into a video that lets the person in the painting say a by the user provided line. First off, the audio of the video must match the provided line, and the face of the painting has to be lip-synced with the produced audio. In addition to the lips and rest of the face must look natural. Using an AI driven system, one of such talking painting, can be produced.

C. RESEARCH

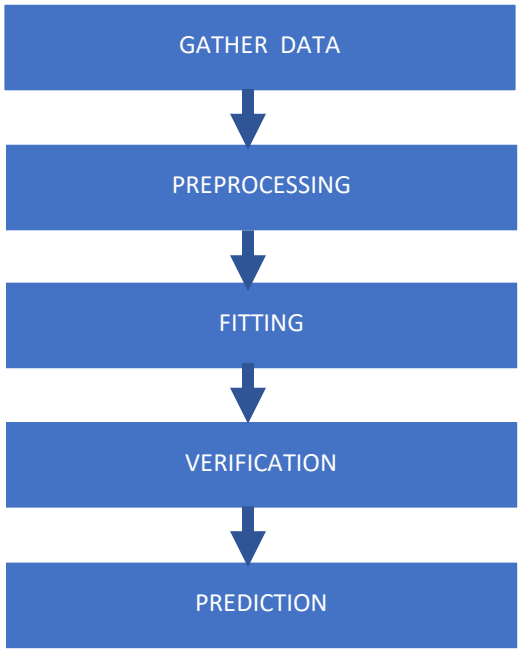
1. MACHINE LEARNING

1.1. WHAT IS MACHINE LEARNING

Classical programming techniques give the system a rigid set of instructions that are carried out in order exactly as stated. Machine learning breaks from this paradigm by instead supplying the algorithm with a goal. The program will then adapt its internal algorithm to be able to carry out the goal as accurate and as efficient as possible. This means that the algorithm “learns” in a similar way as a biological system or human does. [1]

1.2. BASICS

A machine learning model requires that data is provided ,either by hand or by the system, and learns by slightly adjusting parameters each time a run is done.
Deciding which neural network to use is based on the goal, time and information available.
Neural networks need data to learn and many times the struggle comes into play when trying to find input data. [2] [3]



The process of a machine learning model can be explained in 5 steps.

- The first step is gathering data
- The second step is Preprocessing which includes formatting the data into something the Machine understands.
  Something regularly used for this are data frames which can be seen as a matrix where each element is accessible by calling a category.

- The third step is The fitting stage, often called the learning stage, and is responsible for training the system with the data supplied by the user. It does this by slightly adjusting parameters each run. The exact method used depends on the machine learning algorithm chosen. In the case of Deep fakes this is done with Deep learning but more about this later.

- fourth step is the Verification and will run after the fitting stage to calculate the accuracy of the trained model. We do this by splitting some of the input data. After that the system verifies the model based on the testing data and the parameters provided by the trained model.

- As last and fifth step you have the Prediction stage which will apply the trained parameters of the model on unseen, general data.

The distinct data points that we use in machine learning are called features. Features are independent variables that the algorithm calculates the parameters around. These can be seen as inputs from a data source, as for example an input file, a database or an API. [4]

There are two important kinds of features:

Discrete features, where all data points can only assume a finite predefined set of values. In our case, discrete features will always be categorical. This means we select values from a predefined nominal table. This is in

contrast to continues features, where each data point can select from an unlimited set of values. This always includes numerical values.

There are two different kinds of discrete features, nominal and ordinal features [5]:

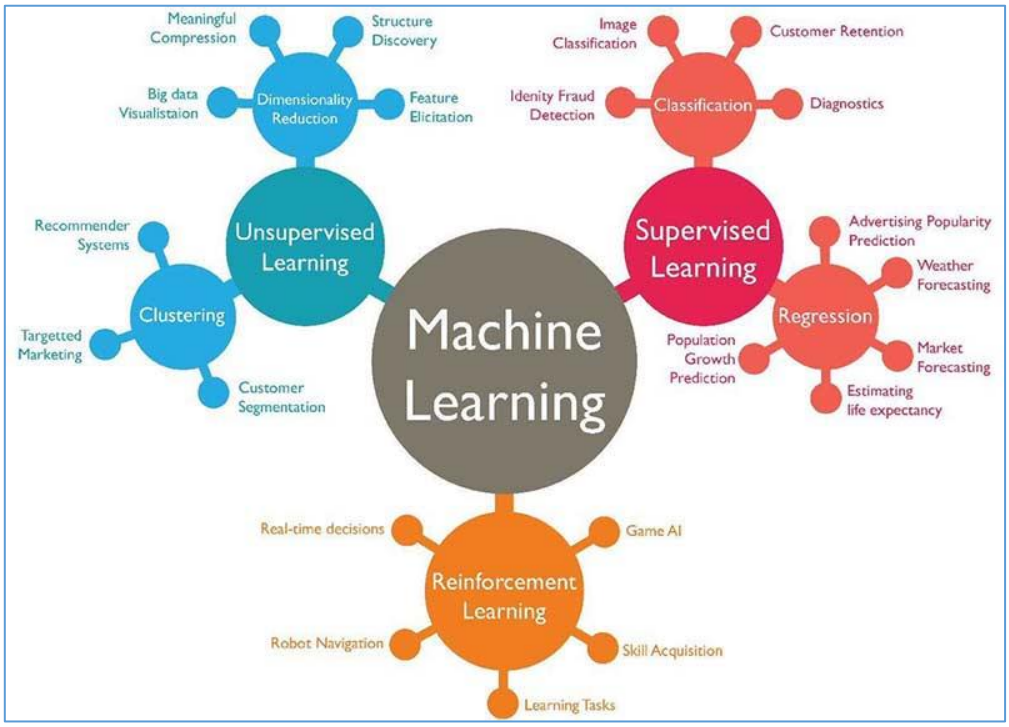
- Nominal values are values that, when our system changes categorial parameters into numbers that it understands, are unordered.
- Ordinal values work exactly the same as nominal values but instead of being unordered these values have a meaning to the order given.

There also exist two kinds of continuous features, ratio and interval:

- Ratio variables are variables like length, or weight, where multiplying with a constant value has a specific meaning. For interval values, like timestamp or temperature, multiplying with a constant does not convey any value. In the practical project we specifically use nominal and ratio features.
- The only interval feature we use is time.

Our machine learning model will use features to optimize internal parameters. We cannot choose how or when our model decides to set or change these internal parameters. This is the essence of machine learning, which separates it from classical algorithms, where all parameters are controlled by the programmer.
In neural networks targets can be seen as rewards. They change depending on the goal of our system and isn't something the user can decide but is still an important process of understanding the principles of a neural network.

1.3. TYPES OF MACHINE LEARNING



Picture 1

There are three major kinds of machine learning models, supervised, unsupervised and reinforced. The main difference between these models is the way the models are rewarded .

Each of them can be further subdivided into continuous and discrete types. [6]

- Accompanying discrete learning is called Classification, continuously supervised learning is called regression. [7]
- Analogue, unattended discrete learning is called Clustering, while unattended continuous learning is called dimensional reduction.

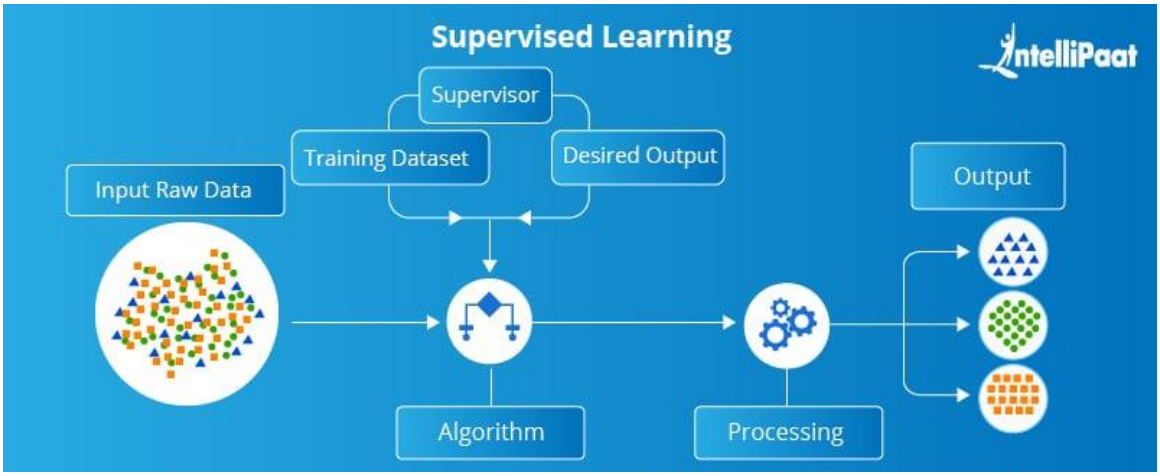
Some other small model types are still applicable, such as genetic learning and graph optimalisation.
Guided learning provides the target with the right data, enhanced learning does not provide a target, but a so-called fitness function that can verify the correctness of the model's results. For unattended learning, no reward function or data is provided.

Shallow learning is the simple optimalisation of variables with a set in stone mathematical model, while in-depth learning makes use of complex neural networks.

Overfitting is the result of an inefficient amount of data and this can result in the system memorizing the data instead of learning it. Underfitting is caused in the same way, but this also means that there is even less data, so that the system cannot learn anything from it.

In both cases, the solution involves adding more data to the system.

1.3.1. SUPERVISED LEARNING:



Picture 2

Supervised learning [8] is a type of machine learning where the data is labeled. Each set of input data comes provided with the exact result the machine learning model should provide. This means that the data will have to be labeled manually by a human, which makes it time consuming and expensive to collect all the necessary data. The flip side of this is that this is a fast and accurate way to train a machine learning model.

If you want to take a look at the algorithms used for this take a look her. [9]

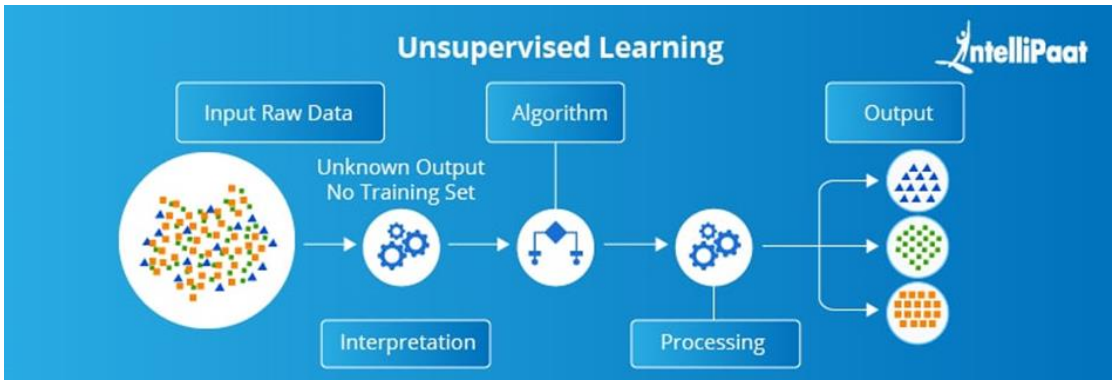
They are 2 types of Supervised learning:

a. *Classification*:
Classification divides data in a finite number of distinct, discreet categories.

b. *Regression*:
Prediction or acquisition of unknown variables in a continuous system.

Supervised learning is mainly used when already labeled data is easily available.

1.3.2. UNSUPERVISED LEARNING



Picture 3

Unsupervised learning [10] is a type of machine learning where the data is unlabeled. The set of input data has not provided any way that the data should be distinguished, and leaves the model completely free to decided how and why to separate the data. The upside of this is that we do not need data that is initially labeled, which makes it easier and cheaper to acquire the data. Another upside is that as the data is not labeled by humans, and thus any biases created by humans will not be present.

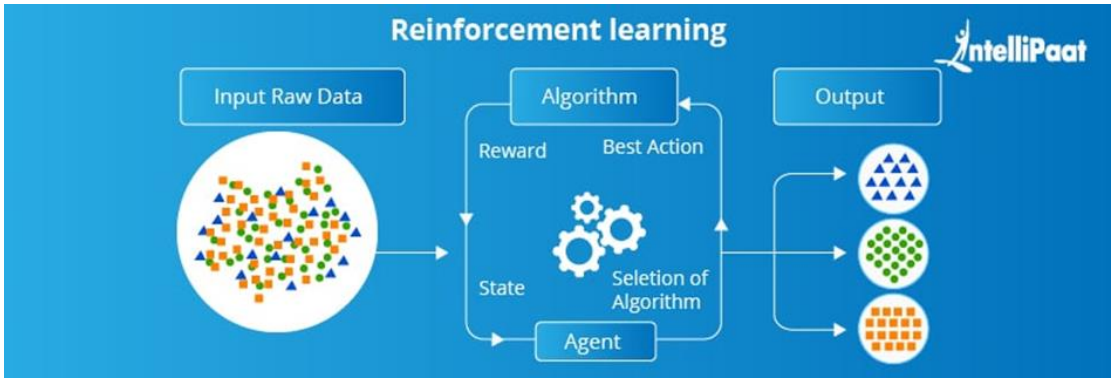
The major downside of this approach is that as the categories are not set in stone, categories will often not be comprehensible by humans, or will make distinctions based on properties that might be unimportant to the task at hand. This model also takes longer to train and verify, due to the increased amount of freedom.

You again have 2 types of unsupervised learning:

a. *Dimensionality reduction [11]*:
This includes going from a higher dimensional space to a lower dimensional space. Thus we reduce the number of relevant features.

b. *Clustering [12]*:
We divide the input data into distinct categories.

1.3.3. REINFORCEMENT LEARNING:



Picture 4

With reinforcement learning [13] instead of using labeled data, we have a reward function (sometimes called the environment) that decides the reward.

Depending on the action the agent performs a positive or negative score will be given.

The environment will notify the agent if it got hit. All scores of the current run will be combined and validated compared to the previous high score. If the score is better we will overwrite it and next run this will be the score we validate on.

1.4. DEEP LEARNING

A neural network consists of a model of layers. These can either be singular or multiple depending on the functionality.

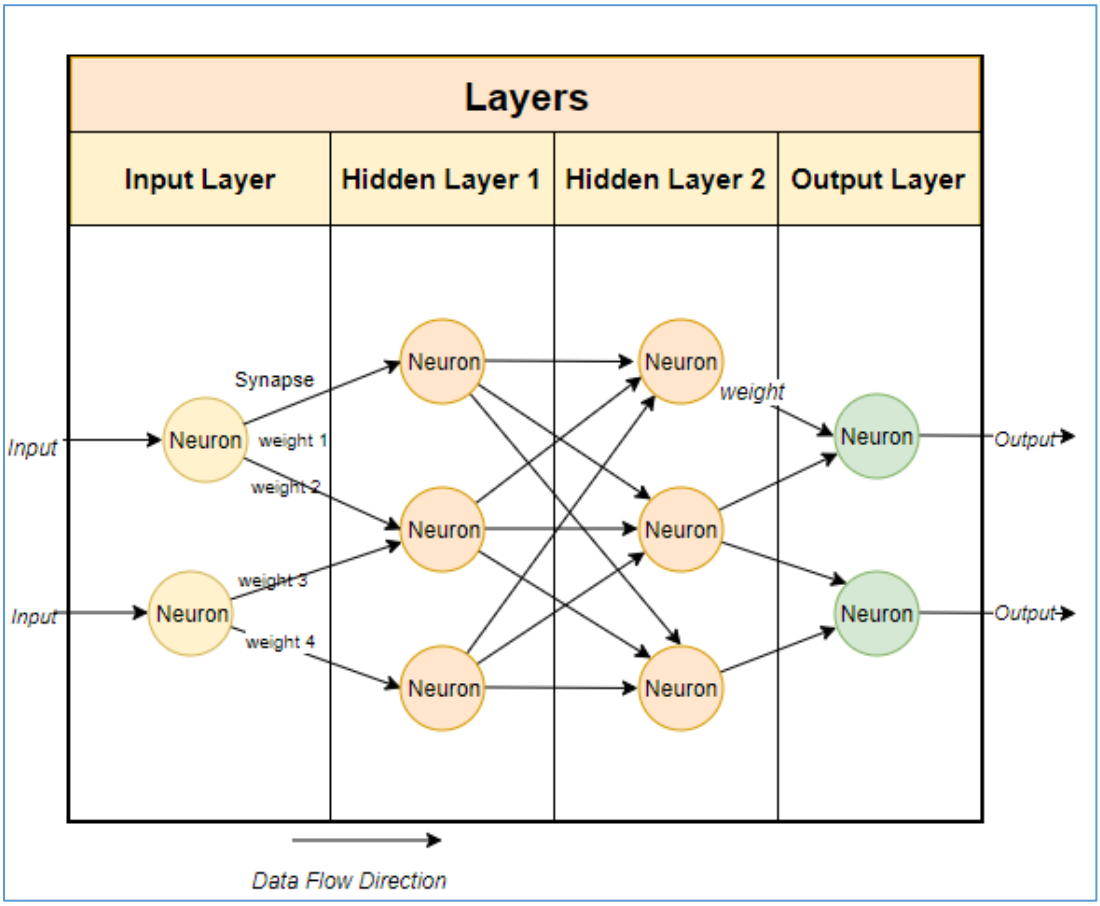
Models with a single layer are called Shallow learning models. they have limited functionality but in a lot of cases this is enough.

Models with multiple layers are called deep learning models. These can become very complex to follow and are the closest resemblance to the human brain [14]

All neural networks in this project will be based on deep learning [15].

1.4.1. LAYERS

Layers [16] are the building blocks of our systems



Picture 5

Layers are self-contained machine learning systems. The first layer of our network takes in the input data, and each layer processes the data gathered from the previous one and passes it on to the next. This makes the full neural network an emergent system that can process data in ways a single layer cannot do. Each layer is build from a set of unconnected neurons who do not communicate with each other, only accept data from neurons from the previous layer and passes it on to the next.

Each layer is responsible for its own functionality and so most of the times a type that is given as input will be constant throughout the layer.

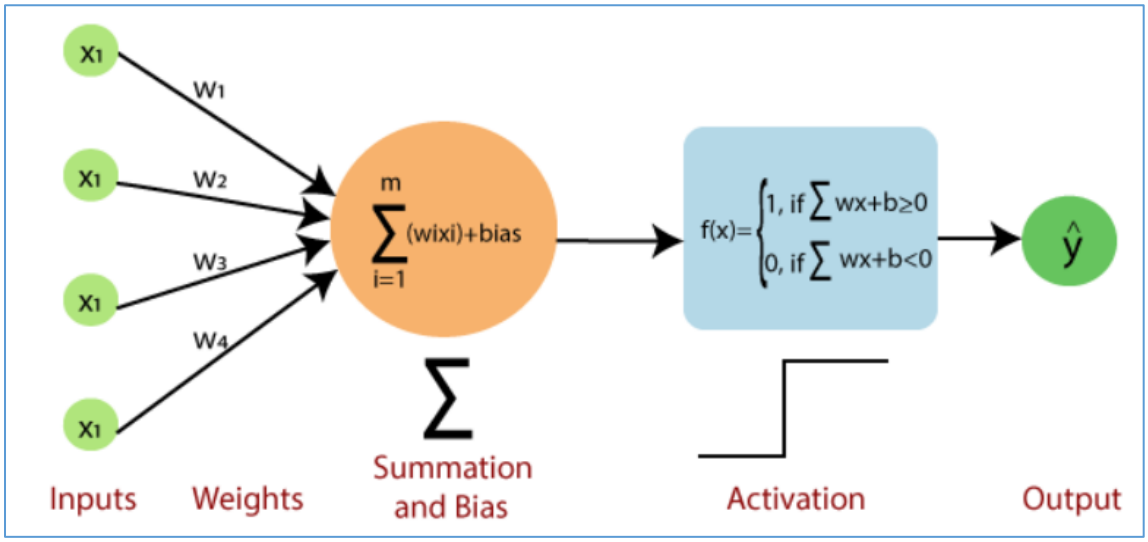
When talking about a neural network the layers between the input layer and output layer are called the hidden layers [17]. These are called hidden because they can't be seen by external systems and the programmer has no access to nor control over these layers.

Further in this paper a more thorough explanation will be given about the specific layers we used during this project.

1.4.2. NEURONS

Neurons [18] also called nodes or units, are the basic cells of our brain. Neurons can be seen as methods and so have limited complexity in terms of functionality. It takes outputs of previous neurons as input, does some simple computation, and then gives us a return value as output.

This way you can create complexity while still keeping the logic of the neurons very simple and clear to understand.



Picture 6

Weights are associated to every input to make sure system based calculations are done correctly. Neurons will get input from different layers. The input values will be combined and an associated weight will be multiplied to them. Furthermore a bias will be added. These weights and biases are chosen by the neuron itself. At the end, an activation function [19] is applied to the specific value. This activation function will be the same for each neuron in the same layer.

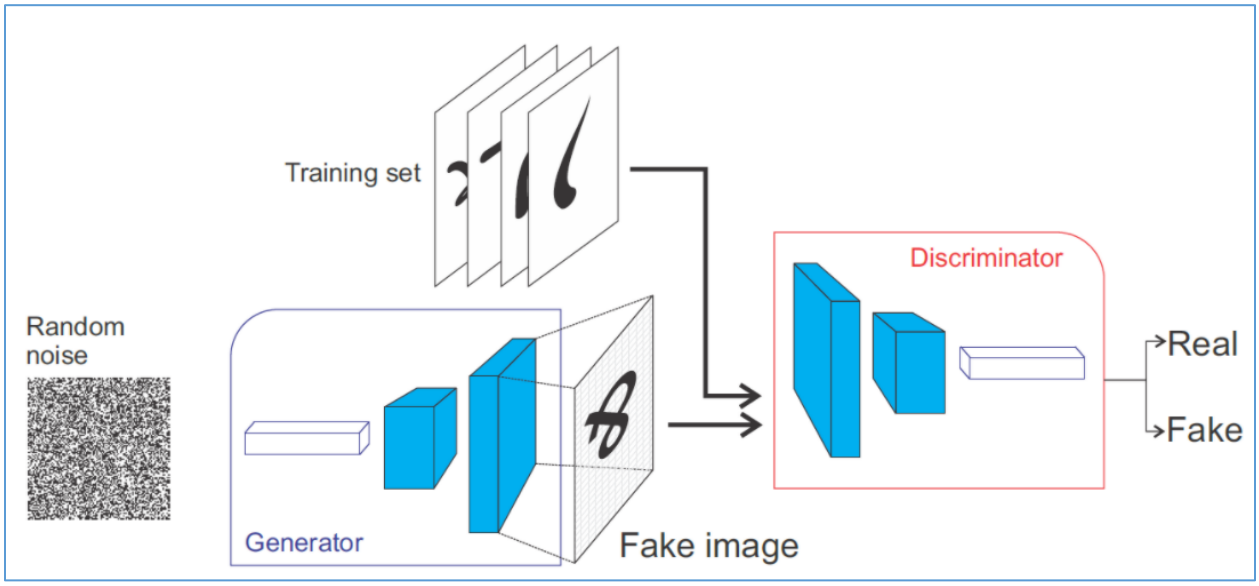
$$Y = \sum (weight * input) + bias$$

In dense layers you have 3 common types of activation functions. These functions are the rectified linear unit (ReLu), Hyperbolic Tangent (tanh) and Sigmoid. ReLu Is usually used where possible since the computations are basic and fast for the system, in fact ReLu is the simplest nonlinear continuous function with are two properties important to make a neural network work.

An important difference between ReLu and the other 2 activation functions is that when we have too many layers problems will occur in terms of accuracy(the vanishing gradient problem) while ReLu doesn't have that problem. Sigmoid and tanh are use more complex computations which their speed a bit slower compared to ReLu. sigmoid is similar to tanh and it is commonly used for classification of binary data in the output layer. Tanh is a hyperbolic function that is used to center data so that the following layers have an easier time learning.

2. GENERATIVE ADVERSERIAL NETWORK

Generative adversarial neural networks [20] are compound neural networks, which means that instead of having a single neural network that learns from functions, it is a coop program where multiple neural networks interact with each other, and compete, improving both the neural networks in that way. This is easiest to do when both neural networks are multi-layered.



Picture 7

A generative adversarial neural network is constructed out of 2 neural networks, a generator and a discriminator.

The generator takes a trained data set and tries to generate data in such a way that the discriminator will interpret it as real.

The discriminator Is responsible for analyzing the data and determining whether or not the supplied data is fake.

When the discriminator improves, it will improve the generator, as the generator gets a reward for misleading the discriminator. If the generator improves, the discriminator will improve because it gets more and more difficult training data. In short, the generator communicates with the discriminator through its output data which is the discriminator's input data. The discriminator communicates back via the reward function, which flows in the opposite direction. This makes the adversarial network a dynamic system that updates itself and adapts itself over and over again, making it an endless cycle of improvement.

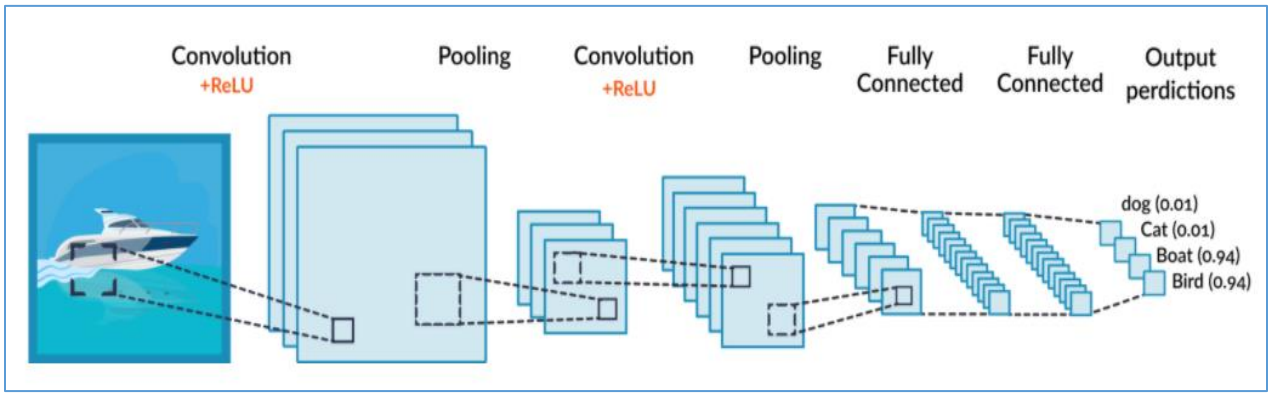
3. CONVOLUTIONAL NETWORKS

A Convolutional network [21][22] or often called CNN or convNet is a type of network commonly used for deep learning and specializes in grid-based data because of the way it is build up. Two dimensional convolutional networks specifically are made for training networks on image data. This is because the structure of the neural network is adapted to the spatial correlation of the pixel positions.

There are many types on convolutional neural networks but only one is covered by this paper which is the 2 dimensional network. a 2- dimensional networks indicates that it is built for two dimensional input, this makes it effective for image processing, as images are two dimensional.

A convolutional network is a sequence of specific layers which are the convolution layer, pooling layer, dense layer and dropout layer. These layers mostly use the ReLu activation function.[23] [24]

An important different between classical neural networks is that instead of matrix multiplication we instead use convolution which makes the convolution layer the core of a convolutional network.



Picture 8

In many cases convolutional layers are stacked. This allows the system to learn more complex features and thus can be easily adjusted when a lower level of features is required. This does however mean that overfitting is a common problem for these layers as we introduce more degrees of freedom.

- A convolutional layer:

  A convolutional layer is given 2 matrices as input, a kernel and the data that needs to be filtered. The kernel is a matrix which includes the learnable parameters of the model. The kernel operates as an analyzer of our data and looks at this data in groups. Each data point is weighted with the relevant value from the kernel and put input the correct group.

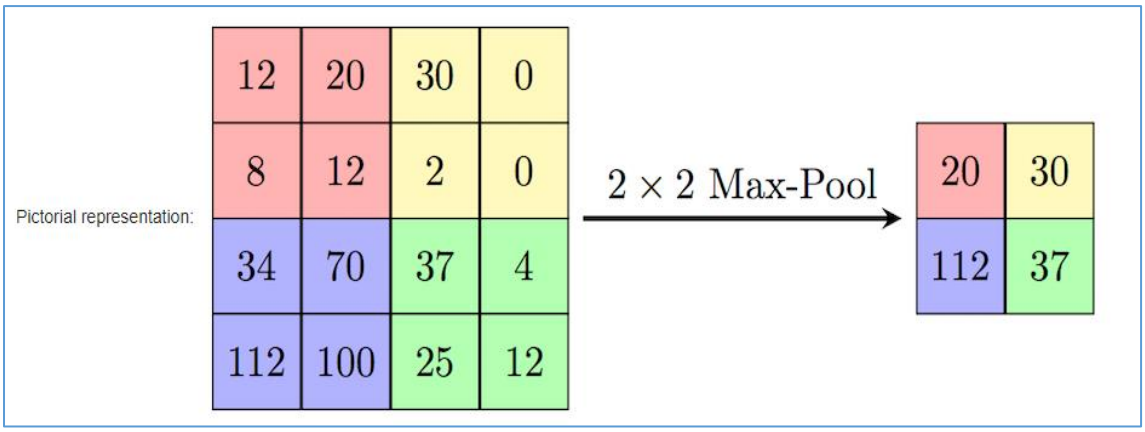
  After this the output of the convolutional layer is given to the pooling layer.

- A pooling layer:

  There exist many types of pooling layers but the only ones important for this project are the average pooling and max pooling layer.

A Max pooling layer indicates that data from the matrix is analyzed and the highest number of all features of the same type is stored in a new matrix. This means that only the most common features are included in the next layer and the other nonexistent functions are forgotten.

Below you can find an example of a 2x2 max pooling layer.



Picture 9

Sometimes you'll also find average pooling layers that take the group average and weighted average based on analyzing how far away the pixel is from the center and take the pixel closest to that average.

- An output layer:
  The output layer [25], or often called the dense layer, is the output layer of our system. In case of a convolutional network this layer is always a fully connected layer which indicates that every individual neuron has a connection to all the neurons of the next layer.

- A dropout Layer:

  As last there is the option to add a dropout layer [26]

  This layer is different from the other layers, in fact it is not a mandatory layer. A dropout layer is a layer that will drop random nodes from our model. In classical programming language, data loss is bad, but in neural networks it can sometimes solve functions that are too close together. In this way, a large part of the impurities is removed, so that the result is actually better.

4. LSTM

Long Short term memory (LSTM) is used in the source code of the lip-sync algorithm and thus is important to understand before moving along [27] [28].

LSTM is a type of RNN (recurrent neural network) which will loop through the data and decide whether it should be remembered or not based on the previous data seen. This is a great way to minimize data before training and can often save costs and time.

RNN is great to use when there will be data that has time attached to it, such as videos or animations.

There are 2 types of RNN's, a short term memory and a long term memory:

- The short term memory indicates that the network will look closely back into the past and decide based on this.
- The long term memory indicates that the network will look far back into the past and will because of this look at the data more broadly.

The data of a face is given to our network and filtered so that only the position of the mouth is returned.

An LSTM is a special case of RNN and is special in terms of being able to learn long term information.

LSTM is a hybrid model of both short-term and long-term memory modules that indicates that some parameters will be remembered for a long time while others change each frame.

This document will not go into too much detail, but for more information on LSTMS [29] the following link is provided.

After this, a video with the lip-sync is made and given to the Pix2Pix network.

5. PIX2PIX

Pix2Pix is a supervised neural network that takes 2 videos as input, one for lip-sync and one for the background. An important thing to be aware of is that both inputs have time associated to them which means Pix2Pix will look at frames and not the full video. Pix2Pix will be responsible for coloring the image to closely resemble the image of the frame.

This model is trained based on pairs of labeled images and then tries to generate the input image into an output image that closely resembles the same frame from the background video.

Pix2Pix is made out of 2 separate networks, the generator and the discriminator.

The generator is training dependent and responsible for generating the image. The input image doesn't have to be related in any way and runs separately with the discriminator.

The discriminator calculates a scalar number which indicates how similar the 2 images are. We will cover this in the next part

As end result you get a new video which contains the merged frames of both videos.

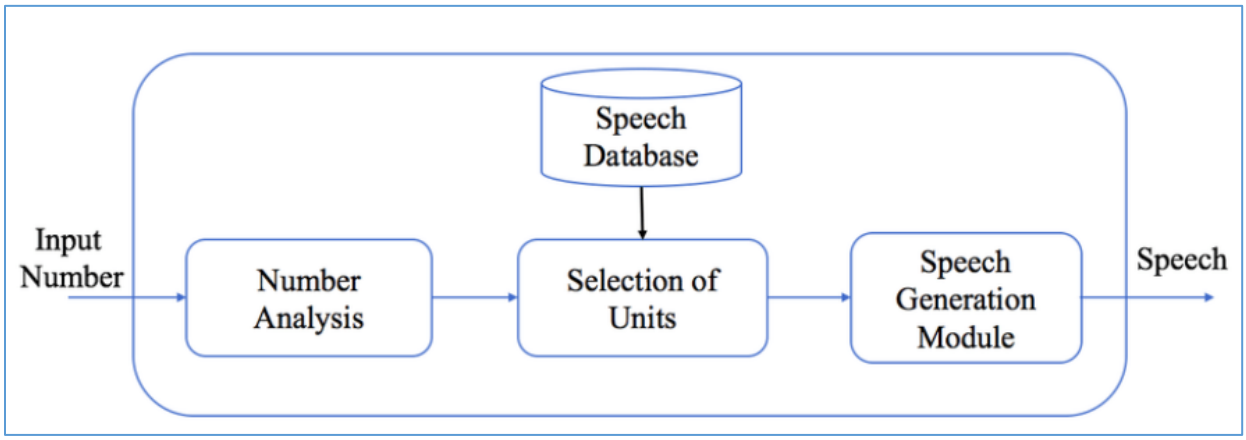
6. PIPELINE

The pipeline consists of 3 separate models, TTS, lip-sync and our deep fake. In This project the choice was made to have 3 different models and put them together into a system that changes text to a talking painting. First a TTS model is needed to change the text into an audio file. Then The lip-sync will change the text into a believable way of speaking and as last this way of speaking will be printed onto the image that needs the animation.

6.1 TTS (TEXT TO SPEECH) [1]

There are a lot of variations of TTS systems but overall they are always developed for specific purposes meaning that a goal should be present before deciding which type is needed. Mainly putting in extra features includes losing something else. This is done by making the module language based or not add intonation to the speech created.

TTS or Text to speech [30] is a method used to turn sentences into audio using supervised learning methods.



Picture 12

A text to speech system usually is build up of multiple parts being the text analysis frontend, an acoustic model and an audio synthesis module which is something that's uncommon for older TTS systems.

First the algorithm will validate the text for any symbols such as numbers or abbreviations and change them into full words.

Then, the frontend generates speech by a sequence to sequence model or in short seq2seq. Using Seq2Seq we are able to link every part of the written text to a phoneme included in the database. This is done by splitting and marking the text so it matches the phonemes of the data.

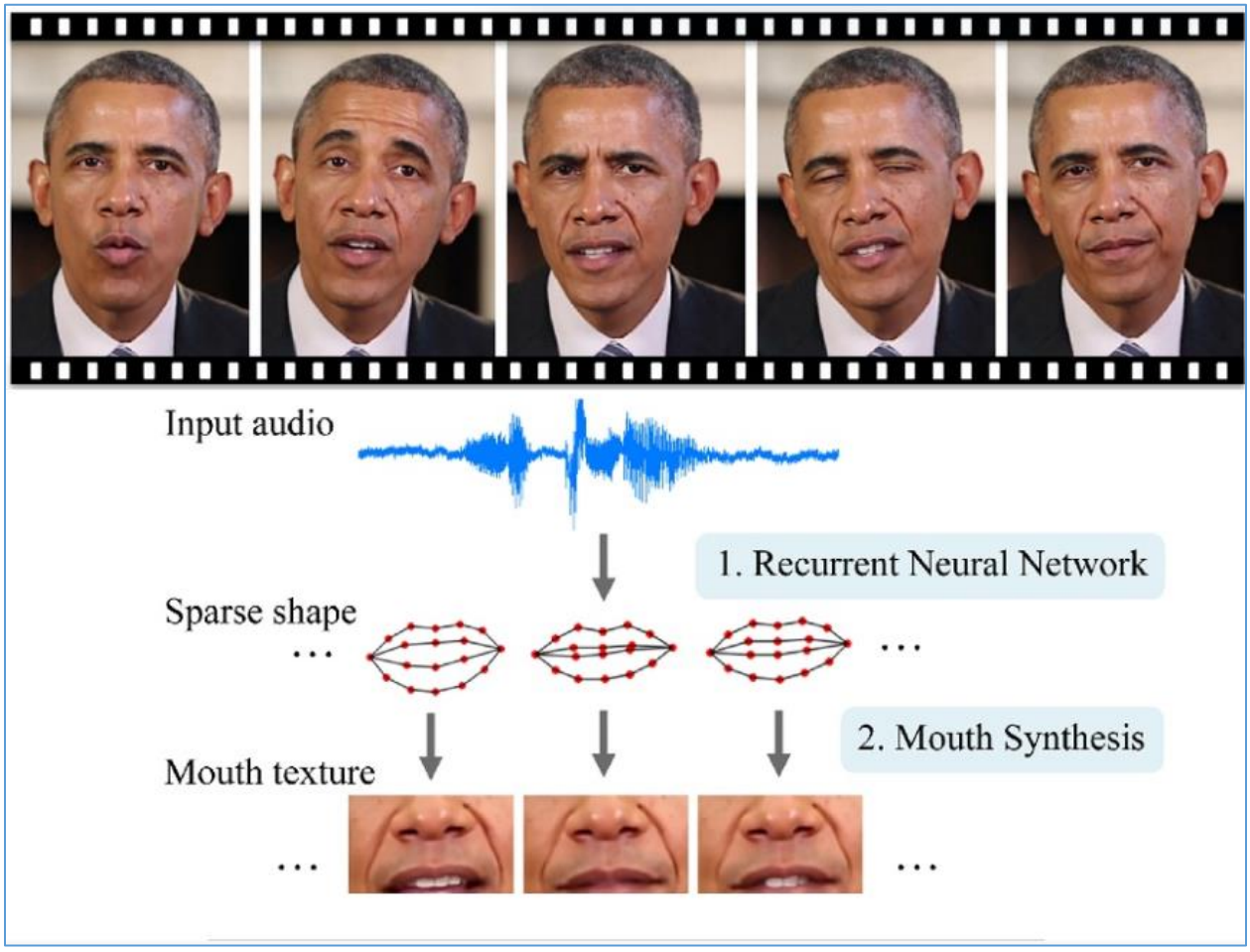
After that, the acoustic model is responsible for Assigning sound to the earlier split words.

A database contains a set of prerecorded voices. The phonemes of these voices will be taken from the database and copied into a full audio sentence.

Finally, an optional module adds volume and intonation to the sentence. The structure of the text will be looked at and change the phonemes where needed. [31]

This will create a wav file which will be used as input for the next module of this project.

6.2 LIPSYNC[2]



Picture 13

Lip-sync [28] is the technique used to convert audio into facial expressions using a trained convolutional network.

- The crated wav file is now used as input for our second network called lip-sync and for this project ObamaNet is used, which uses the supervised learning method. ObamaNet also provides both TTS and lip-sync, but I specifically chose not to use this because I wanted to bring each separate network together.
- A data set of face images is given as input. This will be used for the mouth animations.
- After the TTS, a time-delayed LSTM is used to generate mouth-key points synced to the audio created by our TTS. Key points are specific locations that make up key areas of the face. These key points are for example the mouth, eyes and nose. Once these key points are generated they will be connected forming key areas, in the case of this project only the mouth key points are needed.
- After that the output is used as input in Pix2Pix to generate the video frames conditioned on the key points.
- Pix2Pix returns the positions of the mouth points in a black and blue colored image.
- Animations are then added to the image.

6.3 DEEPFAKE[3]

First order model [32]:

A classical deepfake model needs a source and a target video, and can use deep learning to create a video of the face of the source, with the expressions, thus key point locations, of the target. In this specific case, this approach cannot be followed, because our source, the painting, cannot provide an input video, but only a single image. This means that a different approach has to be used, called a first order model. this is a supervised learning model that uses translation of key points to generate a video based on driving video.

At activation time a source image and a driving video is given. In the case of this project, the source image is the portrait where we want to reflect the lip-sync onto. The driving video is the video provided by the lip-sync model, This video contains all the information we need aside from the background and face. These 2 inputs will be used to generate a video which is the combined version of both.

To do this our model consists of 3 models, the motion estimation, the dense motion and the generator model.

The motion estimation model is based on Monkey-net and includes using transformations to change positions of key points of the source image used. here the key points are given a basic translation and are stored into a set of pairs along with the location of each key point.

These key points are then given to the dense motion model. The motion estimation model is quite limited but it is acceptable for a base. The dense motion model is responsible for fixing this issue. The local approximations are used to calculate the dense motion field of each image. This will give us an occlusion mask which contains the information about whether pixels can be warped or should be replaced by the driving video.

The generator model then takes a set of the key point positions of the motion estimation model and the occlusion mask to create an image. This image will be painted in and warped according the data provided. Once this is done all images are stored in a set and time is associated to them making it into a video file.

D. CASE STUDY

Introduction:

As mentioned in the Pipeline chapter the project consists of 3 parts. These parts were selected based on the open source code available and on the packages used.

Implementations:

By using different open source models the problem of compatibility arised meaning that different models used different versions of the same software. Docker was used to fix this issue. Docker is a containerization system. It runs every service in a separate so called container. The point of containers is that each container acts like a separate virtual machine, so every piece of installed software inside of the container cannot and will not affect other parts in the host system. This means that there will be way less bugs and errors, and thus debugging time is seriously reduced. Containers are built from images, where data from the host system is copied into the container, and runs there.

Each of the different services that are described in the pipeline was put in a separate docker. For the TTS and the deep fake, docker images where readily available. Although the deep fake needed some minor configuration. The docker for ObamaNet was built from scratch, but this was done by translating the installation instructions from the readme into a docker file, such that ObamaNet was installed inside a python docker in a similar way one would install it in a native operating system.

The communication between the different dockers is done with a bash script. The TTS docker works like a web server, so we can easily just start the docker, perform calls to the webservice with Curl, and save the result. The other two dockers do not do this, so the input file have to be copied directly into the docker, and after the docker is finished, the output files have to be copied back out of it. The bash script just copies each file in the right location, such that the three services run consecutively.

At the end, the result of the deep fake is a soundless video, so we use ffmpeg to recombine our video with the sound produced by TTS.

Results:

In the end, we have produced a program that can produce an image of Mona Lisa that says any voice line of choice, and is sort of lip-synced, and has natural movements.

Discussion:

Some problems that came to light were the TTS model and the ObamaNet model. ObamaNet already has a building TTS but the choice was made to implement another TTS for the sake of being able to change the voice if desired. This takes the problem of training the lip-sync of Obama with it. ObamaNet is trained on the voice of Obama specifically meaning that since a different TTS was used the performance and quality of the lip-sync got reduced.

The deep fake however works very well. It is clearly visible that Mona Lisa takes on the movements of Obama. TTS works as expected. We can clearly hear the voice generated is robotic but in terms of quality it is what was expected from an open source TTS system.

Conclusion:

A program was build that creates talking painting, using three open source systems, TTS, lip-sync and deep fake. Our program has successfully produced a talking painting. The sound still sounds a bit robotic. The lip-sync happens, but can be improved upon. Despite this, the movement of Mona Lisa looks natural and works.

E. REFERENCES

[1] DATALYA, machine learning vs traditional programming https://datalya.com/blog/machine-learning/machine-learning-vs-traditional-programming-paradigm, March 5th, 2020.

[2] Intellipaat tutorial neural networks, Neural Network Tutorial - Artificial Intelligence Tutorial (intellipaat.com), 28 June 2019.

[3] UPGRAD, neural network tutorial, Neural Network Tutorial: Step-By-Step Guide for Beginners | upGrad, blog NOV 20 2019.

[4] CHRISTOPH MOLNAR, Interpretable machine learning https://christophm.github.io/interpretable-ml-book/cnn-features.html, 2021-01-18

[5] Bart Steenbergen, Difference between discreet and continuous data http://www.raamstijn.nl/eenblogjeom/index.php/lean-six-sigma/3436-lss-soorten-variabelen-data, 30 March 2018.

[6] Vishakha Jha, types of machine learning, Machine Learning Algorithm - Backbone of emerging technologies (techleer.com), 17 July 2017.

[7] Vishakha Jha, what is regression, Interpretable machine learning algorithm - Linear regression (techleer.com), 21 June 2017

[8] Intellipaat, what is supervised learning, https://intellipaat.com/blog/what-is-supervised-learning/ 20 July 2020.

[9] Rich A Caruana, Alexandru Niculescu-Mizil, An empirical comparison of supervised learning algorithms https://dl.acm.org/doi/abs/10.1145/1143844.1143865, June 2006

[10] Thomas Wood, what is unsupervised learning, https://deepai.org/machine-learning-glossary-and-terms/unsupervised-learning , 08/13/2020.

[11] Ali Ghodsi, Dimensionality Reduction A Short Tutorial, tutorial_stat890.pdf (uwaterloo.ca), 2006.

[12] Wikipedia-bijdragers, "Clusteranalyse," *Wikipedia, de vrije encyclopedie,* https://nl.wikipedia.org/w/index.php?title=Clusteranalyse&oldid=54805654, last change October 19, 2019.

[13] Błażej Osiński and Konrad Budek, what is reinforcement learning, https://deepsense.ai/what-is-reinforcement-learning-the-complete-guide/ , July 5, 2020.

[14] Vsauce, The Stilwell Brain, https://www.youtube.com/watch?v=rA5qnZUXcqo, 12 dec. 2018.

[15] missinglink.AI neural networks in more detail, Complete Guide to Artificial Neural Network Concepts & Models (missinglink.ai), 2019

[16] Tim Dettmers, NVIDIA Accelerated Computing, Deep Learning in a Nutshell: Core Concepts | NVIDIA Developer Blog, November 3, 2015.

[17] Farhad Malik, What Are Hidden Layers?. Important Topic To Understand When… | by Farhad Malik | FinTechExplained | Medium, May 20, 2019.

[18] Stacey Ronaghan, Deep Learning: Overview of Neurons and Activation Functions [ ]| by Stacey Ronaghan | Medium, July 26, 2018.

[19] MissingLink.ai, 7 types of activation functions: how to choose, https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/, 2019

[20] Ian J. Goodfellow , Jean Pouget-Abadie , Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair , Aaron Courville, Yoshua Bengio , Generative Adversarial nets, Departement d'informatique et de recherche op ´ erationnelle ,Generative Adversarial Nets (nips.cc), 2014

[21] Mayank Mishra, Convolutional neural networks explained, Towards data science, Convolutional Neural Networks, Explained | by Mayank Mishra | Towards Data Science, August 26 2020.

[22] Wikipedia contributors, "Convolutional neural network," *Wikipedia, The Free Encyclopedia,* https://en.wikipedia.org/w/index.php?title=Convolutional_neural_network&oldid=1000906936, last change December 24, 2020.

[23] Jason Brownlee, A gentle introduction to the rectified linear unit (ReLu), Machine Learning Mastery, A Gentle Introduction to the Rectified Linear Unit (ReLU) (machinelearningmastery.com), August 20 2020.

[24] Anonymous, Why do we use Relu in neural networks and how do we use it, Stackexchange, Why do we use ReLU in neural networks and how do we use it? - Cross Validated (stackexchange.com), Nov 8 2018

[25] Jan-Willem Middelburg, Artificial Neural Networks, Cybiant, Artificial Neural Networks , May 2nd, 2019.

[26] Jason Brownlee, A gentle introduction to dropout for regularizing deep neural networks, Machine Learning Mastery, https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/, August 6, 2019.

[27] Wikipedia contributors, "Long short-term memory," *Wikipedia, The Free Encyclopedia,* https://en.wikipedia.org/w/index.php?title=Long_short_term_memory&oldid=999574811, last change November 17, 2020.

[28] Yuyu Xu, Andrew W. Feng, Stacy Marsella, Ari Shapiro, A Practical and Configurable Lip Sync Method for Games, USC Institute for Creative Technologies, MIG2013_lipsynch.dvi (usc.edu), 2013

[29] Christopher Olah, Understanding LSTM networks, https://colah.github.io/posts/2015-08-Understanding-LSTMs/, August 27, 2015.

[30] Wikipedia contributors, "Speech synthesis," *Wikipedia, The Free Encyclopedia,* https://en.wikipedia.org/w/index.php?title=Speech_synthesis&oldid=1000955031, last change December 4, 2020.

[31] Google, Creative Commons Attribution 4.0 License, Apache 2.0 License, Cloud text to speech basics, https://cloud.google.com/text-to-speech/docs/basics, 16/11/2020.

[32] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, Nicu Sebe , First Order Motion Model for Image Animation, https://papers.nips.cc/paper/2019/file/31c0b36aef265d9221af80872ceb62f9-Paper.pdf, 2019.

F. PICTURES

Picture 1 Vishakha Jha, Machine Learning Algorithm - Backbone of emerging technologies, Machine Learning Algorithm - Backbone of emerging technologies , https://www.techleer.com/articles/203-machine-learning-algorithm-backbone-of-emerging-technologies/,July 17, 2017.

Picture 2, 3 & 4 Intellipaat, Supervised vs Unsupervised vs Reinforcement Learning | Intellipaat, Updated on 26th Dec, 2019.

Picture 5 Farhad Malik, Understanding Neural Network Neurons | by Farhad Malik | FinTechExplained | Medium, May 18, 2019.

Picture 6 Judy Nduati, Section, Introduction to Neural Networks | Section, October 25, 2020.

Picture 7 Rezaul Karim, Generative adversarial networks, Oreilly, Generative adversarial networks - Java Deep Learning Projects [Book] (oreilly.com), 2021

Picture 8 MissingLink.ai, Convolutional Neural Network Tutorial: From Basic to Advanced - MissingLink.ai, 2019

Picture 9 Mr. MacKenty, Max-pooling / Pooling - Computer Science Wiki, last edited on 27 February 2018.

Picture 10 Monica Mundada, Sangramsing Nathusing Kayte, Pradip Das, Implementation of Concatenation Technique for Low Resource Text-To-Speech System Based on Marathi Talking Calculator, Block diagram of unit selection text-to-speech (TTS) synthesis. After [29]. | Download Scientific Diagram (researchgate.net), 29-31 August 2018.

Picture 11 Luke Jones, Fake-News Threat: Nearly Perfect Lip-Sync between Audio and Unrelated Video Shown by Researchers - WinBuzzer, July 12, 2017.

G. SOURCE CODE

(1) Eren Gölge, Mozilla / TTS, Mozilla, https://github.com/mozilla/TTS ,last edited December 22 2020

(2) Karan Vivek Bhargava, ObamaNet, https://github.com/karanvivekbhargava/obamanet,last edited December 18 2020

(3) AliaksandrSiarohin , First order Model, https://github.com/AliaksandrSiarohin/first-order-model, last edited December 24 2020