Journal of Theoretical and Applied Information Technology
31st March 2021. Vol.99. No 6

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3D MODELLING BY MEANS OF ARTIFICIAL INTELLIGENCE

BEBESHKO B. [1], KHOROLSKA K. [2], KOTENKO N. [3], DESIATKO A. [4], SAUANOVA K. [5], SAGYNDYKOVA S. [6], TYSHCHENKO D. [7]

[1,2,3,4,7] Kyiv National University of Trade and Economics, Department of Software Engineering and Cybersecurity, Kyiv, Ukraine,
[5,6] Almaty University of Power Engineering and Telecommunications, Almaty, Kazakhstan

E-mail: [1]thismushroom@gmail.com, [2]k.khorolska@knute.edu.ua, [3]kotenkono@knute.edu.ua, [4]desyatko@gmail.com, [5]sauanovaklara@gmail.com, [6]tomka2001@mail.ru, [7]tyfran@ukr.net

ABSTRACT

For years humanity has developed some solid perception of the world around them. According to this perception, it is easy to describe complex structures through a short explanation. For example, by telling a person to "think of a yellow submarine, flying in the sky", the person in question would have an exact image of the machine you are talking about, without having seen it ever before. Along with this - artificial intelligence becomes a rapidly growing area of research, and it is possible that artificial intelligence will make it possible for computers to gain perception of the world. Artificial intelligence can be used to solve problems without the need to specify how to solve the settled task. This paper outlines features of 2D images recognition and the creation of 3D models, using AI and machine learning accordingly. Therefore, it would make designers work much easier, - graphic designers and architects would be able to only specific features instead of spending a lot of time on the actual drawing. Moreover, if models were computationally generated, the developing process would become much more simple and could result in a higher developer's performance rate. The key purpose of this work is to define the possible limitations of CNN usage for 3D models generation, taking into account output resolution and generation swiftness.

Keywords: *Artificial Intelligence, Neural Network, 3D, 2D, Image, Models, Graphics*

1. INTRODUCTION

For centuries humans have developed some solid perception of the world around them. According to this perception, it is easy to describe complex structures through a short explanation. For example, by telling a person to "think of a yellow submarine, flying in the sky", the person in question would have an exact image of the machine you are talking about, without having seen it ever before. Along with this - artificial intelligence (AI) is a rapidly growing area of research, and it is possible that AI will make it possible for computers to gain perception of the world [1]. Such an ability would enable computers to perform complex tasks, without someone managing its single action. Therefore, it would make designers work much easier, - graphic designers and architects would be able to only specify features instead of spending a lot of time on the actual drawing. Machine learning (ML) is a section of artificial intelligence, it is used to make systems automatically learn from experience instead of being programmed exactly how to perform a specific job [2]. Therefore, it can be used to solve problems without the need to specify how to solve the settled task. This paper outlines features of 2D images recognition and the creation of 3D models, using AI and machine learning accordingly.

The creation of 3D models is a complex and time-consuming job for everyone who has little to pour experience in 3D modeling. This results in a big issue for game engines since developers either have to learn 3D modeling, hire a specific 3D designer or purchase third-party models. Therefore, if models were computationally generated, the developing process would become much more simple and could result in a higher developers' performance rate. To produce the computational 3D modeling usable, there are three fundamental requirements:

1. Satisfying Quality.
2. Rapid generation time.
3. Simple in use.

The aim of artificial intelligence is to make the interaction and communication between human and AI systems more natural [19]. As it is known- the human brains vision seems to be very easy functioning. It does not take any difficulty to tell apart a dog and a cat, read a word or recognize a human face. But in difference from humans - these tasks are really difficult problems for solving with a computer. Recognition process only seems easy because the human brain is really good at perception and as a result in understanding images. During the last years, machine learning has made great progress in solving these difficult problems. In particular, model called - deep convolutional neural network can result in reasonable performance on solving difficult visual recognition tasks which are matching or exceeding human performance in some aspects [20].

Few publications [3, 4] states that it is possible to generate a 3D model, using a single input image and convolutional neural networks (CNN) (Figure 1). One can assume that it can be considered as simple enough for the third consumer-friendly requirement. The assumption that requires testing is the limitation of the output resolution and swiftness of generation time, which has not been ever specified in any of the papers mentioned above.

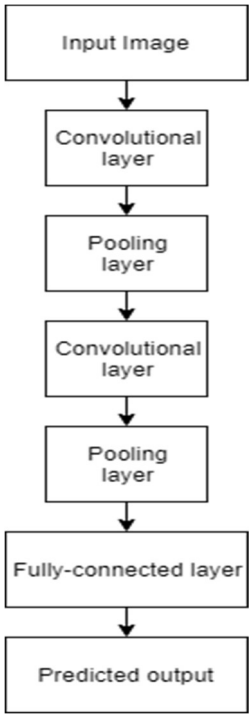

Figure 1. Visual Representation of the Convolutional Neural Network Algorithm.

The key purpose of this work is to define the possible limitations of CNN usage for 3D models generation, taking into account output resolution and generation swiftness. Moreover, this work also focuses on balancing, between an increase in the resolution quality and time consumption for output generation. According to all the above mentioned the research question can be stated as: "How generation speed can be impacted according to output resolution while generating 3D models from 2D images with means of convolutional neural networks approach?"

As it was mentioned above - this project solution focuses on three areas (Satisfying Quality, Swiftness and Simplicity) to make the 3D generation program satisfying for common use. Therefore, in turn to get an answer to the research question of this work, the following objectives must be accomplished:

1. Figure out a machine learning model to extract data from a 2D image.
2. Figure out a generative 3D machine learning model.
3. Develop and cross-link both machine learning models.
4. Test and adjust parameters to achieve both best output quality results as well as best time consumption results.
5. Achieve a conclusion in form of the final 3D model. When the project of this work is complete, the result should give one the ability to perform 2D to 3D transformation of an image into the model. The image should be taken from a mobile camera and further converted to a 3D model.

2. THE AIM OF THE ARTICLE

The aim of this scientific paper is to figure out the limitations of using CNN for 3D voxelized model generation, taking into account the performance characteristics of the algorithm. Underperformance characteristics, one considers the following as the most valuable: generation speed and output resolution.

Moreover, this paper will also heavily focus on the possible scale of the trade-off, between improving the resolution against the increase in time for generating output. Therefore, one of the aims is to obtain the most efficient result which will be considered as final.

The results from this research work focus on three sub-areas, which are: quality, speed, and simplicity. These sub-areas are the most valuable to make the 3D generation tool customer friendly.

When this project is complete, the aim is to enable a 2D-3D conversion of an object, from a

hand drawing a picture to a 3D object. The object should be able to be imported into a 3d editor tool like Blender and a game engine like Unity 3D and Unreal Engine.

3. LITERATURE REVIEW

As it was stated before - this research work aims to classify and extract data from 2D images with further conversion into 3D models. This work has been heavily inspired by recent progress in several different areas in the field of artificial intelligence and machine learning.

Since 3D modeling claims a wide application in a range of areas it has, therefore, drawn much attention in recent years. However, there are not many completed pieces of research in this field. Most of the existing works are aiming to describe the classification and building process of the 3D model from the premade or in other words existing initial data. Nevertheless, for the creation of a 3D model from a 2D image one should discover possible manipulations with the 3D model itself and therefore in this area, there was a range of good research works.

In the scientific paper [23] authors presented a neural network approach for the classification and creation of the existing 3D prismatic parts. Such an approach is based on a 3D part being modeled by the initiator of its three projected views and then converted to input vectors for a polygon creator. Nevertheless, such an approach considers only the initiator information of the 3D model projections. The overall distinguishing ability of this method is way far from a production-ready solution.

In the research works [24, 25] authors presented a machine learning methodology for the classification models. The approach of enhanced shape distribution has been utilized to convert a 3D mesh into a range of histograms, and therefore the kNN algorithm was chosen. As it became obvious from the study, runtime parameters of the algorithm have been tuned in the course of the training phase. The performance of the kNN initiator has been demonstrated by the range of experiments, but the success rates in the shown results were, however, not satisfactory.

As for the 2D image classification and data extraction where also several scientific works.

In [26] authors used the Caltech 101 set that was among the first standardized datasets for multi-category image classification, it contains 101 object classes with generally 30 training sample images per single class. Later the set was reworked by [27] and became Caltech 256, which is the set with the increased number of object classes to 256 and added images with greater scale and background variability. However, since all of the mentioned data sets have not been manually verified, they contain many errors that are in a way making it unsuitable for precise algorithm evaluation activities.

The most suitable for precise algorithm evaluation activities and, therefore, for ILSVRC became the PASCAL VOC dataset proposed in [28]. This dataset gives a standardized test object for object detection, image classification, object segmentation, person layout, and action classification. Most of the choices in ILSVRC have been provided by the influence of the PASCAL VOC. Therefore, ILSVRC amplifies PASCAL VOC's goal of standardized training and evaluation processes of the recognition algorithms by more than an order of magnitude in the number of object classes and images: PASCAL VOC has 20 object classes and 21,738 images compared to ILSVRC with 1000 object classes and 1,431,167 labeled images.

There is a range of datasets with standardized online evaluation algorithm compained similar to ILSVRC: the aforementioned PASCAL VOC [28], Labeled Faces in the Wild [29] for unconstrained face recognition, Reconstruction meets Recognition [17, 18, 30] for 3D reconstruction and KITTI [31] for computer vision in autonomous driving.

Concerning 2D to 3D conversion one of the most advanced works was Deep3d [36]. Main idea behind Deep3d was a usage of a neural network for prediction of disparity map and using disparity map for further generation of the right view of image from the left view. Nevertheless, Deep3d image generation models are not fully differentiable. Another disadvantage of Deep3d was that the size of the image is constantly fixed. In work [36], the author describes this by predicting a distribution of a range of disparity values at each single pixel, instead of predicting a each single value. The final correct image is further obtained by computing the expected value at each pixel in the right image over the range of disparity values. In [37] the authors proposed a fully convolutional deep neural network loosely inspired by the supervised DispNet architecture described in [38]. They consider depth estimation tasks as an image reconstruction problem. Therefore, their architecture can solve tasks for the disparity field without requiring ground truth depth since learning is unsupervised. Authors are

using a left-right consistency check to improve the precision of synthesized depth images. Given a single image "A" at test time, their goal is to learn a function (f) that can predict the per-pixel scene depth, ˆd=f(I). Specifically, during the training period, they obtain access to two images Il and Ir , corresponding to the left and right color images from a calibrated stereo pair, captured at the same moment in time. They attempt to find the dense correspondence field d r that, when applied to the left image, would enable to reconstruct the right image.

Our article proposes a more simple method based on the existing previous experience from related works.

4. METHODOLOGY

The main approach in this paper project is in use of the empirical development method. Which states that some bigger problems should be split into smaller tasks using reductionism. Therefore, the generative model is first split into two separate neural networks. The first one is aimed to extract data from the input image, the second one is aimed to utilize the output from the first neural network in order to perform 3D model creation. Both of these neural networks were trained and optimized separately using an iterative, empirical, exploratory method. Small parameters or dataset changes are made to the neural network to be retrained to see if the neural network is reducing the mean squared error (MSE).

The scientific name of the neural network training process is called machine learning. It is a process that uses statistics and probability on a set of data to do classifications. One can compare ML against traditional programming and define that the common programming utilizes data together with a program to generate a specific output. But, by switching places of the output and program it would become a machine learning model, that produces a program instead of the same output as common programming. The input is 'data' and 'output' if a program that can generate similar output from another set of data.

There are several different choices of approaches when working with machine learning (Fig. 2).

They are supervised approach (widely called - "learning with the teacher"), unsupervised approach, semi-supervised approach and reinforcement learning approach. A supervised learning approach can be used and usually is used when both input and output are accessible to the developer. Its aim is to find a mapping between the input and output, this mapping can then be further used on unseen input, to map it against a similar output. An example is a classifier that can classify a web-form submit as 'submitted by bot' or 'submitted by human'. If the developer has a dataset, consisting of classified web-submit actions, the dataset can use the behavior within the web-form as input and the classification as output. This could create a 'program' that can tell the difference between bot and human.

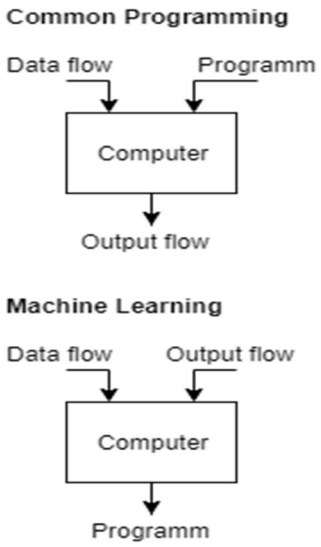

Figure 2. Example of Common Programming and Machine Learning Construction Principles.

The unsupervised learning approach is widely used to cluster data. An example can be shown as a process of definition on how any language was constructed, along with verbs, nouns or adjectives. An unsupervised learning approach could also be used by providing to a model a lot of text-driven data, and specify that the model should group sentences that are used in a similar way.

Whenever some input was specified in prior as well as some output, but not all output one can use an approach that is a bit more complex - called Generative Adversarial Network (Figure 3). It is an approach of two neural networks working against each other. For example, one creates a program that can act like a bot in web-form and two neural network units that compete against each other. One tries to act like a bot and the other tries to tell if the web-form submit is produced by a bot from the other neural network, or if it's a real web-form submit from the correct dataset. While this process would train, both networks would improve, resulting in 'better' bot-like acting and a better bot detecting. This also can be considered

as a semi-supervised model since some data was existing from the start but new data was also created during the process of learning.

There also may be cases whenever neither input nor output was given, but the data about possible actions and a current state of an environment are available. In such a case, a reinforcement learning approach could be implemented. It starts by randomly choosing different possible actions and by making assumptions on if it was a good action, depending on the environment changes after the action was executed. For example, it can be described by the process of a helicopter to fly as high as possible, it would quickly learn that the rotor blade on the tail does not give as much height as if the big propeller blade on top of the helicopter is rotating.

A best-practice way of machine learning usage is to use neural networks which are a bio-like programming paradigm. Which is to say a technique that gives computers some power to learn from observed data.

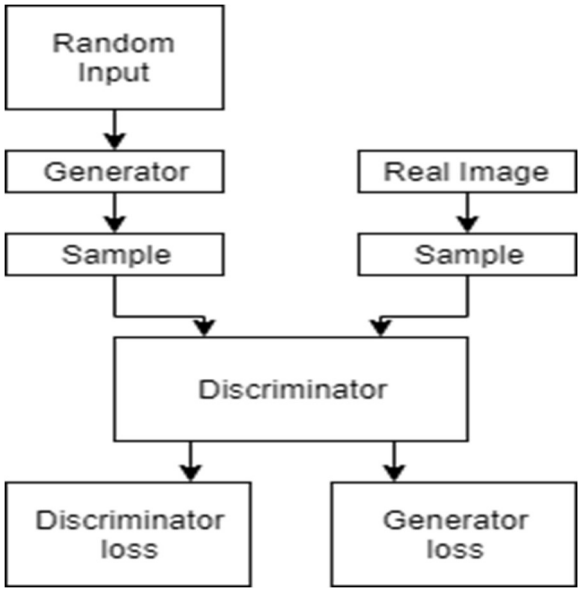

Figure 3. Visual Representation of the Generative Adversarial Network

This paper considers two methods for 3D model creation from image. The first one is Generative Adversarial Network and the second one is Autoencoder (Figure 4) based on octree structured architecture. The datasets used in this paper for training purposes were taken from an IKEA public product dataset [5], which in its own order was analyzed manually for labeling and corrupted data removal for further use.

A convolutional neural network (CNN) is a sort of network inspired by the visual cortex [6], inherited from the multilayer perceptron (MLPs) [7]. CNN is a sort of mechanism that can be used to represent data with little parameters, like classification. To explain the pros of CNN, there is a public resource named Large Scale Visual Recognition Challenge (LSVRC) [8]. Since the middle of 2013, this service has held annual challenges in computer science. During this challenge, people around the world try to develop artificial intelligence that is able to correctly classify most images within fifteen datasets called ImageNet. Nowadays, neural networks proposed in the challenge are able to reach an error rate of 3.1%, comparing the same few years ago with a rate of 20%. Such results really show neural network effectiveness in production use. Therefore, CNN is a new solid standard of image classification and recognition. CNN works by bypassing filters through the input data and performing a dot product calculation between the filter and the data, the result of this task generates a new so-called - activation map. The activation map itself can be processed to become an output or it can become a part of the next hidden layer. In turn, the hidden layer would be processed in the same or similar way as the previous layer.

Anyways convolutional neural networks (CNN) outperform any possible humans' recognition rate. However, such systems should be continuously manually improved. Another problem with such systems is that they require

accurate data to be trained before they are actually being deployed. It is essential that the system is fast enough to recognize image and that the training should be accomplished without much difficulty and also be fast [20, 33].

The convolutional layer always consists of three parts: convolution itself, activation function and pooling. Dot product multiplication between filter/weights and input layer or hidden layer. Activation function makes the neural network nonlinear, to make it able to find more complex unobvious patterns. Pooling is aimed to reduce the data, in order to remove unnecessarily created neurons and therefore improve the performance.

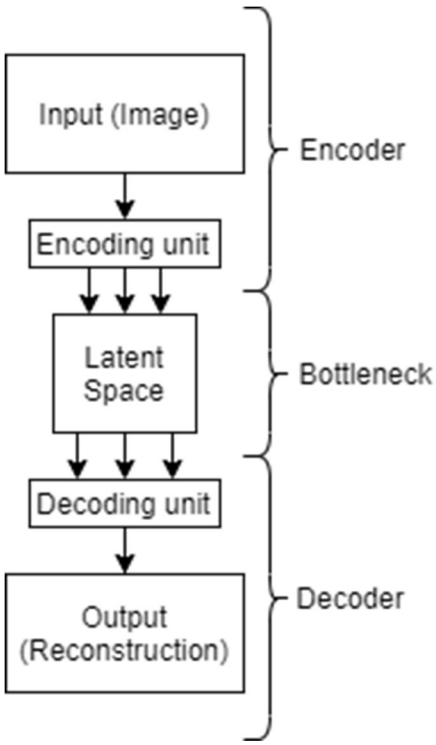

Figure 4. Visualization of the Autoencoder Training Structure.

The aim of the Activation Function is to make the neural network nonlinear. In turn, it makes it possible to define more complex structures within datasets.

Pooling is aimed to reduce the actual size of each activation map, to improve the speed of the neural network. A downside of pooling is that it will remove the opportunity of the neural networks to define the location of models on the image since only the "best" value is saved. Two widely used pooling functions are max-pooling and average-pooling. Commonly a $2x2$ sized window slides over the matrix of the dataset and if max-pooling is used, only the highest value in the window is stored. Mean pooling will store the average of all values in the window. By having a $2x2$ sized window the size of the activation map can be reduced by as much as 75%.

The additional loss functions compare a ground truth answer and generated output. It is what decides how well the model is performing, during training the loss function is minimized each iteration in order to improve the model. A few normally used loss functions that will be shown and used are the Cross-entropy loss, Mean Squared Error (MSE).

Cross-entropy is calculated by adding up the log value of the correct prediction times -1 to achieve a negative value. For example, the model that should process and define the color of a scarf, if the output is 80% chance that is purple and 20% that it is yellow, therefore the ground truth would be purple the cross-entropy would be $-log(0.8) \approx 0.11$ which is a lesser value than if the scarf would have been yellow ($-log(0.2) \approx 0.69$). $L1$ is calculated by adding the absolute difference of output and ground truth. By using the same example as above, with a small difference that the output is how much purple is in the scarf according to the 'R' in 'RGB' which has an interval of $0 - 255$. If the model would suggest 245 for the Red value and the ground truth is 250, this would give an error of $(245 - 250) = 5$. $L2$ is calculated by adding up the squared difference between the output and ground truth. By using the same example as above in $L1$, the calculated error would be $(245 - 250)\,2 = 25$. As shown, $L2$ will give a higher failure for a bigger error. Mean Squared Error (MSE) is calculated similarly, the difference being that it will divide the answer to get an average value. $L1$ & $L2$ are similar to each other, the big difference between them.

Optimization Algorithm is aimed to minimize or maximize the loss function. It is processed by calculating the gradient, which is used to decide if the weights and biases should be increased or decreased. Generally, there are 3 optimization approaches that will be considered in this work: Gradient Descent (See Figure 5), Adam, and Adagrad. They were chosen because of the fact that they have been proved to provide a satisfying result in other similar projects, and they are the most popular optimization approaches [9]. Gradient Descent has been chosen since it was seen as the foundation of optimization approaches in this area. Adaptive Moment Estimation, known as Adam is a popular approach which calculates the learning rate for each weight/bias. It performs

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

well in practice due to its swift converge ability and good learning speed. Adagrad is an adaptive gradient approach, each iteration of which will decrease the self-learning rate which in theory will make it do smaller modifications the longer it trains, to easier find a minimum

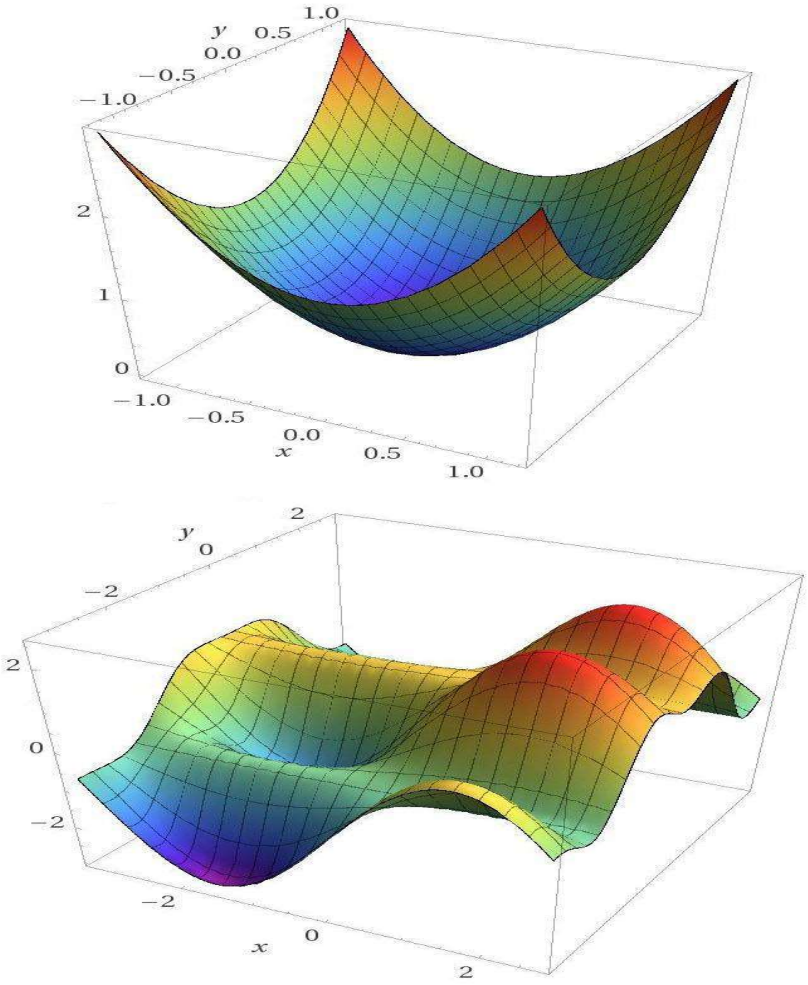

Figure 5. Visual Presentation Of The Gradient Descent Algorithm.

Autoencoder is one of the most popular uses of CNN. Basically, it is usually known for its ability to learn using an easy to understand, unsupervised learning method. It utilizes convolutional networks to reduce the data value of a given object into a latent space. It will further use this latent space to transform the original object. This proves that an autoencoder is able to extract data from an object in order to transform it. To improve the autoencoder it will reduce the difference between the original and the transformed object, with backpropagation. Which main structure consists of an encoder and a decoder. The purpose of the encoder is to reduce the amount of data needed to describe an object, then the decoder will try to transform the object.

Generative AI had a huge breakthrough in 2014, once presented a new neural network model called Generative Adversarial Network (GAN) [10, 32]. The main concept in Generative Adversarial Network is in further flow: instead of training one neural network against predefined answers (learning with the teacher), this model consists of two neural networks, a Discriminator, and a Generator. The Discriminator-network's aim is to correctly classify whether an input data is part of a dataset or not. The Generator-network's aim is to fool the discriminator by creating data that it will

classify as "part of a dataset". By training both networks against each other, the Discriminator will become better at identifying objects from the dataset and the Generator will become better at creating data with attributes that match the real dataset.

Data is one of the most important aspects of dealing with artificial intelligence. To reduce the scope of this work, the dataset is only looking into tables that were presented in the IKEA dataset [5]. The IKEA dataset is chosen due to its free-to-use and that it is relatively easy to modify and change. To make sure that no corrupted data is failing the neural network, multiple datasets are considered, using both existing data and generating a new dataset. The three approaches described below are only used to modify the input, the 3D output is the same for all image dataset. 3D model files from the IKEA dataset are converted into a voxelized python-format, using a Blender 3D editor. It uses a simple GUI that is easy to understand and the local computer's GPU which makes the conversion quick and seamless.

The solution of this work will use MSE to gain a quantitative data-driven comparison between ground truth models and the output. The neural network was improved by reducing the MSE in prior.

5. IMPLEMENTATION

Deep convolutional neural networks (CNNs) have achieved significant improvements in different vision tasks, including classification, detection and segmentation. However, the increasing model size and computation makes it difficult to implement DNNs on embedded systems with limited hardware resources [16, 35].

For the purpose of reproducibility, this part starts with the environmental setup, followed by the architecture of the CNN (Fig. 6).

The software and hardware used in this work are the following:

1. For the experiment shown in this paper was used Python 3.5, but also in future experiments we plan to use C++ in case for better possible performance due to possible low level core approaches implementation.
2. Tensorflow
3. Windows 10
4. GPU: NVIDIA GeForce RTX 2080 8GB
5. 32 GB RAM
6. Dataset Size: input images ~ 4500, output objects ~ 20

The basic architecture was made of two convolutional neural networks (CNN), the first one was called 'Convolutional Autoencoder' and the second one was called 'Convolutional 3D Decoder'. The job of the first CNN is to extract data from an image, this data is further transferred as input to the second CNN. The second CNN job is to generate an octree structured vox space from the latent space. This process is repeated several times in a row in order to increase the resolution of the voxel-space. Due to the 3D model generator being harder to generalize, there were data extractions tasks to make the latent space similar to the same model, independent of the angle of the model in the input image.

The data extraction is heavily based on the same approach as in J. Wu work 2016 [11]. The main and only single difference is in the dataset. In order to make it more common in the recognition of models from different angles, all images of the same object are transformed to the same output. Therefore, it will also make it easier for the 3D generative neural network since the angle of the input is irrelevant for a 3D model.

Generating 3D models is not a trivial task in common sense, thus two different approaches were taken into consideration. Conditional Generative Adversarial Network was the first approach, due to its was widely heard in the area in content generation and is proved to give a satisfying result [3]. The second approach is a conditional autoencoder that uses an octree structure. Early in the development of the Conditional Generative Adversarial Network, it was shown that it had exponential growth in memory usage while increasing the resolution. This is not a stable solution, thus the approach with conditional autoencoder was focused.

The idea behind conditional autoencoder structure was to make the neural network generate a model in steps. The foundation is inspired by a sculpturing process, which in this case starts with a single voxel that is split into smaller pieces that are being validated to remove unnecessary voxel/space, which in turn results in enhancing the resolution. Each step was trained individually, which enabled the neural network to be able to train on creating a high resolution as possible. The first step generated $4x4x4$ (21 voxels), for instance, this step could find the proportions of the model. The model that is twice as high as width would only need to use 64 voxels (one side of the cube). To understand the power behind this, let one assume that one would like to create an object with the quality of $128x128x128$ (x, y, z) voxels. In this case, the first step would discard half of the voxels $(128 * 128 * 128 / 2 = 1\,048\,576$

voxels permutations), which is a lot. Therefore, each iteration of the generation could improve the quality of a generated chair.

The total amount of voxels required to represent a table in the training dataset, corresponding to a $128x128x128$ quality, is between ~ 5000 and ~ 70000 voxels. Comparing the worst case (~ 70.000) against mutations for each voxel in the space, would reduce the total number of predicted voxels to ~ 70000 from 1 048 576 (~ 6.7%) and the best case is ~ 5000 predictions down from 1 048 576 (~ 0.47%). The fact is that the percentage of the total voxel-space decreases by each LoD, therefore much more gain from increased quality. The result is that the percentage of the total voxel-space decreases by each Level of Detail, percentage of total voxel used in Table 1 for different Level of Detail.

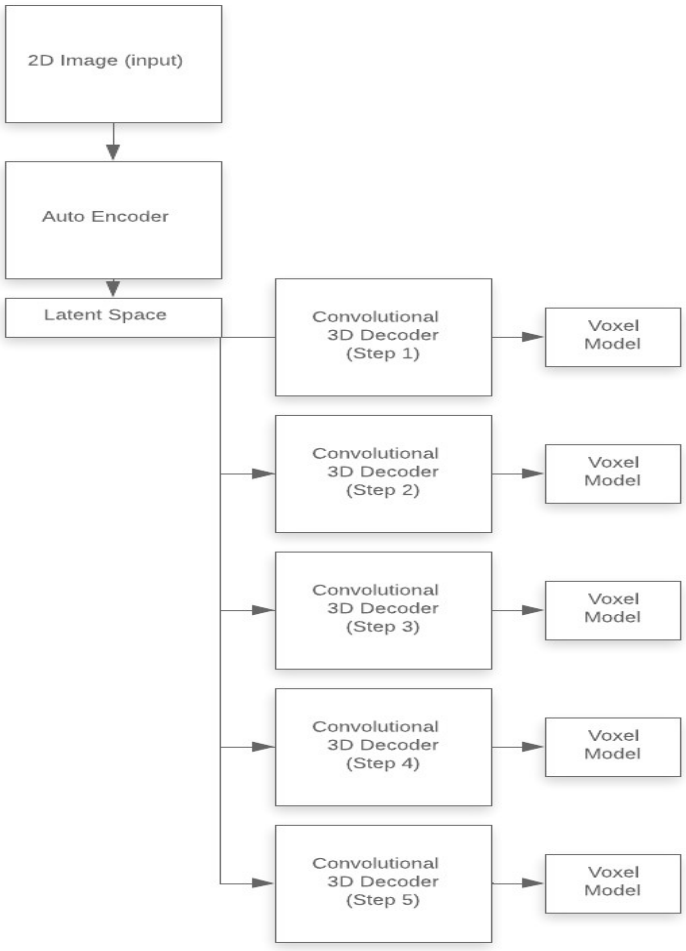

Figure 6: Architectural Structure of Neural Networks Used During Generation

Table 1: Percentage of Voxels Used for Each Level of Detail

Level of Detail	Height / Width / Depth	Percentage of used voxels
0	2/2/2	37.5 − 100%
1	4/4/4	21.4 − 81.8%
2	8/8/8	10.2 − 46.2%

3	16/16/16	4.2 − 22.1%
4	32/32/32	2.2 − 12.5%
5	64/64/64	1.1 − 6.5%
6	128/128/128	0.5 − 3.3%
7	256/256/256	0.1 − 1.8%

The result from different iterations of training the 3D generation CNN were gathered from the process of definition of the optimal dropout, activation function and loss function. The results were improved as the dropout was increased, due to the generalization part being in the data extraction, this was an expected result. The result is presented in Table 2.

Table 2: MSE from the Empirical Approach to Optimize the Drop

Drop	0.5	0.75	0.9	0.95	0.99	1
Loss	0.25177	0.1588	0.04324	0.0283	0.0281	0.0042663

The testing of different setups for activation functions was performed by training each setup on 500 epochs. The tests use the same activation function for the input layer and all hidden layers, the output layer may differ. Leaky ReLU for hidden layers and Tanh for the output layer gave the lowest result shown in Table 3.

Table 3: MSE Result After 500 Training Epochs, Using Different Activation Functions

Hidden AF	Output AF	Loss
ReLU	None	0.2457
ReLU	ReLU	0.2986
ReLU	Sigmoid	0.05748
ReLU	Tanh	0.15502
lReLU	None	0.162
lReLU	lReLU	0.0586
lReLU	Sigmoid	0.0261
lReLU	Tanh	0.00656
lReLU	Softplus	0.00912

6. FURTHER RESEARCH

Further research and development in the described field will be divided into two separate branches.

The first branch of further research will be sharpened on the system and in particular on the algorithm modification so that it can handle a higher Level of Detail. The current approach may handle only the 7th level with states for $256x256x256$ xyz-samples. A further plan is to increase the supported Level of Detail to 9th. This means that it will have the ability to handle $512x512x512$ and $1024x1024x1024$ xyz-samples respectively. However, such an approach may lead to an increase in the drop rate. Therefore,

the algorithm should be modified to minimize that increase.

The second branch is more practical and an "ecosystem" building branch. The main idea of this branch is to combine all tools that are used in one solid package and integrate it with some engines like Unity3D and Unreal Engine, which were primary game engines but in light of modern days deeply become more universal tools offering not only game creation but all sorts of environment simulation. Both the Unreal and Unity engines support 2D and fully 3D rendered productions. They both run the latest technologies including Volumetric lighting, Physically-Based Rendering (PBR), Post Processing, Global Illumination (GI), Advanced shaders and many more. Although developers might be able to produce similar results using either engine, this is unfortunately an area where Unity falls behind. Unity has been more suitable for 2D and 3D development, whereas from the onset Unreal has been focusing more on graphics for 3D development.

Also another way is using clustering methods to build a neural network. Clustering technique is one of the most important tools for knowledge discovery, during which the samples are divided into categories whose members are similar to each other. One of the most common and widely-used clustering solutions is partition-based clustering algorithms such as K-Means and K-Medoids which have attracted a lot of attention in the field of customer clustering [21]. Image data as a text data is the most common form compared to other types of data stored on search engines when searching for a specific element among a collection of large amounts of documents. There are several steps that precede image retrieval, the most important one is image clustering. Image clustering is a process of organizing documents by dividing them into several separate groups, each group contains documents with similarly related topics and completely different from the other groups. It plays an important role in data mining applications such as knowledge discovery, pattern recognition, and information retrieval [22, 34].

7. CONCLUSIONS

By looking at the result, it is possible to assume a future where a similar system could be used for 3D model generation. The key idea of this work was to gain knowledge about different problem areas for generating 3D models from 2D images. However, with the current implementation, the output is too poor and slow for high resolution.

How significant is the trade between generation swiftness and output resolution when generating 3D models from a 2D image, using convolutional neural networks? The trade is in our favor. Gained resolution grows exponentially by each Level of Detail, whilst the generation time is not linear. Since the swiftness has a correlation to the amount of mutation needed to move from one Level of Detail to another, and the mutation needed is getting a lower percentage value by each increasing Level of Detail.

In conclusion, increasing the resolution also increases effectiveness. However, the total number of mutations was still growing rapidly, which in turn means that even though a higher quality is technically more effective due to the trade, it is still not viable to generate high-resolution models, since the total generation time will get too high.

An ethical aspect that is considered for this work was to be protective of the training data. It should not be easy to recreate the original training dataset if it consists of sensitive data. According to the United Nations Sustainable Development Goal (SDG) 8 (Economic growth), it is important to encourage and help with the formalization of small-sized enterprises.

This project could give smaller businesses a better chance in the gaming industry since they might not have the starting capital to hire a graphical team from the very beginning. With artificial intelligence that can help with creating 3D environments, it would be possible to create games more quickly and give smaller businesses a better initial position.

REFERENCES:

[1] Karpathy A. (2016). "Generative Model", Openai.

[2] "What is Machine Learning? A definition", Expert systems, 2018, http://www.expertsystem.com, last accessed 2020/05/19

[3] Wu J., Zhang C., Xue T., Freeman W. T., Tenenbaum J. B. Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling, *In NIPS, pages 82–90, 2016.*

[4] Häne C., Tulsiani S., Malik J. (2017). Hierarchical Surface Prediction for 3D Object Reconstruction

[5] Lim J. J., Pirsiavash H., Torralba A. (2013). Parsing IKEA models: Fine pose estimation, *IEEE International Conference on Computer Vision*

[6] Deshpande A. (2016). A Beginner's Guide To Understanding Convolutional Neural Networks

[7] Kafunah J., Backpropagation in Convolutional Neural Networks, *arXiv:1709.05804v1 [cs.LG] 18 Sep 2017*

[8] Russakovsky, O., Deng, J., Su, H. et al. ImageNet Large Scale Visual Recognition Challenge. Int J Comput Vis 115, 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y

[9] Walia A. S., Types of Optimization Algorithms used in Neural Networks and Ways, 2017

[10] Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., and Bengio Y., "Generative adversarial nets", *Advances in Neural Information Processing Systems 27, Montreal, Quebec, Canada, 2014, pp. 2672-2680.*

[11] Popescu A. C. and Farid H., "Exposing digital forgeries by detecting traces of resampling," *IEEE Transactions on signal processing, vol. 53, no. 2, pp. 758–767, 2005.*

[12] Qian Y., Dong J., Wang W., and Tan T., "Deep learning for steganalysis via convolutional neural networks." *Media Watermarking, Security, and Forensics, vol. 9409, pp. 94 090J–94 090J, 2015.*

[13] M. Lin, Q. Chen, and S. Yan (2014). "Network in network", *International Conference on Learning Representations*

[14] Cire¸san D.C., Meier U., Masci J., Gambardella L.M., and Schmidhuber J.. High-performance neural networks for visual object classification. *Arxiv preprint arXiv:1102.0183, 2011.*

[15] Hinton G. E. (2012). "ImageNet classification with deep convolutional neural networks," *Advances in neural information processing systems, pp. 1097–1105.*

[16] Nguyen H. (2020). A Lightweight and Efficient Deep Convolutional Neural Network Based on Depthwise Dilated Separable Convolution. *Journal of Theoretical and Applied Information Technology, Vol. 98. Iss. 15, pp. 2937-2947*

[17] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

[18] Khalifa, Nour Eldeen M., Mohamed Loey , and Mohamed Hamed N. Taha. "Insect pests recognition based on deep transfer learning models." *Journal of Theoretical and Applied Information Technology 98, no. 01 (2020).*

[19] Mostafa A. H., El-Sayed H., Belal M. (2020). Benchmarking of convolutional neural networks for facial expressions recognition. *Journal of Theoretical and Applied Information Technology, Vol. 98. Iss. 18, pp. 3104-3115*

[20] Kryvoruchko, O., Bebeshko, B., Khorolska, K., Desiatko, A., Kotenko, N. (2020). Artificial intelligence face recognition for authentication. *Technical Sciences and Technologies, 2 (20), 139-148.*

[21] Mousavi S., Boroujeni F.Z., Aryanmehr (2020). Improving customer clustering by optimal selection of cluster centroids in k-means and k-medoids algorithms. *Journal of Theoretical and Applied Information Technology, Vol. 98. Iss. 18, pp. 3807-3814*

[22] Gabralla L., Chiroma H. (2020). Deep learning for document clustering: a survey, taxonomy and research trend, *Journal of Theoretical and Applied Information Technology, Vol. 98. Iss. 22, pp. 3602-3634*

[23] Wu, M.C., Jen, S.R., 1996. A neural network approach to the classification of 3D prismatic parts. Int. J. Adv. Manuf. Technol., 11(5):325-335. [doi:10.1007/BF01845691]

[24] Ip, C.Y., Regli, W.C., 2005a. Content-based classification of CAD models with supervised learning. *Comput. Aided Des. Appl., 2(5):609-617*

[25] Ip, C.Y., Regli, W.C., Sieger, L., et al., 2003. Automated learning of model classifications. *Proc. 8th ACM Symp. on Solid Modeling and Applications, p.322-327. [doi:10. 1145/781606.781659]*

[26] Fei-Fei, L., Fergus, R., & Perona, P. (2004). Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *In 2004 Conference on Computer Vision and Pattern Recognition Workshop, CVPRW 2004 (January ed.). [1384978] (IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops; Vol. 2004-January, No. January). IEEE Computer Society. https://doi.org/10.1109/CVPR.2004.383*

[27] Griffin, Gregory and Holub, Alex and Perona, Pietro (2007) Caltech-256 Object Category Dataset. *California Institute of Technology . https://resolver.caltech.edu/CaltechAUTHORS:CNS-TR-2007-001*

[28] Everingham, M., Van Gool, L., Williams, C.K.I. et al. The PASCAL Visual Object Classes (VOC) Challenge. *Int J Comput Vis.*

88, pp. 303–338 (2010). https://doi.org/10.1007/s11263-009-0275-4

[29] Huang G. B., Ramesh M., Berg T., Learned-Miller E., Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. *University of Massachusetts, Amherst, Technical Report 07-49, October 2007*.

[30] Urtasun, R., Fergus, R., Hoiem, D., Torralba, A., Geiger, A., Lenz, P., Silberman, N., Xiao, J., and Fidler, S. (2013-2014). Reconstruction meets recognition challenge. *http://ttic.uchicago.edu/ ~rurtasun/rmrc/*

[31] Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR).*

[32] Radford A., Metz L., and Chintala S., "Unsupervised representation learning with deep convolutional generative adversarial networks, " *arXiv: 1511. 06434, 2015.*

[33] Akhmetov, Berik & Tereykovsky, I. & Doszhanova, A. & Tereykovskaya, L.. (2018). Determination of input parameters of the neural network model, intended for phoneme recognition of a voice signal in the systems of distance learning. International Journal of Electronics and Telecommunications. 64. 425-432. 10.24425/123541.

[34] Tereikovska, Liudmyla and Tereikovskyi, Ihor and Mussiraliyeva, Shynar and Akhmed, Gulmaral and Beketova, Aiman and Sambetbayeva, Aizhan, Recognition of Emotions by Facial Geometry Using a Capsule Neural Network (September 20, 2019). *International Journal of Civil Engineering and Technology, 10(3), 2019,pp. 1424-1434, Available at SSRN: https://ssrn.com/abstract=3457106*

[35] J. C. Batista, V. Albiero, O. R. P. Bellon and L. Silva, "AUMPNet: Simultaneous Action Units Detection and Intensity Estimation on Multipose Facial Images Using a Single Convolutional Neural Network," *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), Washington, DC, 2017, pp. 866-871, doi: 10.1109/FG.2017.111.*

[36] JunyuanXie, Ross Girshick, Ali Farhadi jxie@cs.washington.edu,ross.girshick@gmail.com, ali@cs.washington.edu ,Deep3D: Fully Automatic 2D-to-3D Video Conversion with Deep Convolutional Neural Networks University of Washington.

[37] Clment Godard Oisin Mac Aodha Gabriel J. Brostow,Unsupervised Monocular Depth Estimation with Left-Right Consistency University College London.CVPR 2017

[38] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovit- ¨ skiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In CVPR, 2016. 2, 3, 4, 5