MODELS4ARTIST: AN INTELLIGENT POSE-BASED IMAGE SEARCH ENGINE TO ASSIST ARTIST CREATION USING ARTIFICIAL INTELLIGENCE AND POST ESTIMATE

HuiBing Xie[1], Yu Sun[2]

[1]Northwood High School, 4515 Portola Pkwy, Irvine, CA 92620
[2]California State Polytechnic University, Pomona,
CA, 91768, Irvine, CA 92620

ABSTRACT

Since some years ago, the popularity of drawing has been increasing. There are a lot of existing tools to help people to improve their drawing [5]. Some tools provide human body images, so people can practice their human body drawing [6]. However, users cannot find the desirable pose images since these tools provide only a list of images but it cannot be sorted by pose. Thus, we proposed a tool in which users can move the joints of a stick figure to obtain the matching human pose image. In our experiments, the result shows that the engine matches 87% of the human images and the stick figure. Also, we performed data analysis with feedback from 10 high school students. The result shows that 5 out of 10 students were satisfied with our tool.

KEYWORDS

MediaPipe, Pose Estimate, Drawing, Matching

1. INTRODUCTION

With the rapid development of modern society, digital art has been gradually introduced into mainstream art forms [7].

This software is designed to help beginners learn to draw. Drawing has always been a very popular art form because its entry threshold is not high. With the rapid development of modern society, digital art has gradually been introduced into mainstream art forms; many digital paint tools have been developed [8]. The convenience of online drawing tools has attracted a large number of people to learn drawing, most of them are youth. Whether it is traditional art or digital art, the basic skills of painting are very important for a painter. However, it takes a lot of time to practice the basic skills of painting. Many people do not have so much time to spend several years practicing the basic drawing skills, but they still want to paint the works in their mind. The software can search for dynamic reference images of what the user wants to draw, allowing even beginners to accurately draw the human body that requires long practice [9].

Some of the model pose techniques and systems that have been proposed to the drawing field, which allows the user to DIY the body model so they have a reference to the pose they want [10]. For example, the Figurosity website provides a variety of images; however, these proposals assume the users already have the knowledge of the joints and bones of the human body, which is rarely the case in practice. Their implementations are also limited in scale, with samples given that the model is very

David C. Wyld et al. (Eds): CSML, NET, BDHI, SIPO, SOEA- 2023
pp. 29-38, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130103

rigid, the user could not reference a precise pose that they want to draw. Other techniques, such as separate reference search for body parts, because separate body parts reference search require the user to have been familiar with the overall relationship of the pose. The method used cannot be too sophisticated and often results in inaccurate search results.

In this paper, we follow the same line of research by pose estimate, which is a technology that uses Artificial Intelligence to analyze human's movements that separate the human body to different parts and use simple lines to represent the human movement [11]. Our goal is to create a useful and simple tool to help correctly draw different human poses for users who want to learn how to draw people.

Compared to the existing Figurosity, our tool provides an interface in which sketching users can move joints of the stick figure and get the desirable human pose that can be useful when they paint or draw. Also, we provide a smart user interface in which users can get the desirable image in less than 1 minute by using Artificial Intelligence tools [12]. In the Figurosity website, they can use filters to search images. However, if users search for specific images, the users would need to search them manually and use a lot of time trying to search for the desirable images. We let users design the pose they want, this gives users more freedom in drawing, and drawing is exactly the kind of art that needs more freedom.

In two experiment scenarios, we demonstrate that our application retrieves the desirable images with the human poses by extracting landmarks of the human body from different images. In the first experiment, we observe two different algorithms to match the human pose of the image and the stick figure pose. For this experiment, we use 15 stick figure poses and search for images that have higher matching scores. Then, we compare the accuracy between both algorithms to observe which of the algorithms perform better. For the other experiment, we do a data analysis to observe the performance of the application in terms of the user interface. We used the feedback from 10 High School students, after they used the application. We provided a questionnaire asking them if the experience using the interface was easy, neutral or complicated and respond if the application provides the desirable images or not.

The rest of the paper is organized as follows: Section 2 gives the details on the challenges that we met during the experiment and designing the sample; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the experiment we did, following by presenting the related work in Section 5. Finally, Section 6 gives the conclusion remarks, as well as pointing out the future work of this project.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Matching Images

The initial design is to let users draw their own posture to match people, according to the user's drawing posture to match the picture. But then it turns out that there's a lot of uncertainty, we do not know what the user is going to draw and we do not know if it's going to match the pose that the user wants. Later the scheme was changed to make a manipulable 2D matchstick figure model with joints that could help the program output more accurate results. Although we used MediaPipe for the finished product, the model of MediaPipe itself has a lot of joints [13]. In order to better imitate the appearance of the painter's drawing, the draft in reality, we spent a lot of time modifying and reducing the number of joints in the model.

2.2. Match a Close Photo with the Match

The main part of the program is to match a close photo with the match person model's pose. We spent a lot of time researching ways to do this, and finally used Mediapipe's technology. Initially, we decided to design an algorithm that could calculate the distance between the joints of the matchstick figure model and then match them. After practice, we found that this method was very inaccurate and often could not get the results we wanted. We decided to change the algorithm, a new one written by calculating the angle between the joints to match the image of the program. The results are better than before, but still not accurate enough. On this basis, we continued to modify and refine the algorithm, not only distinguishing the left and right joints of the match man, but also increasing the proportion of hands and feet, because these two places can best reflect the overall appearance of a posture.

2.3. Python UI => Website- Learn HTML

The program starts as a Python UI. One of the most challenging parts of the process is to make the program web-based. We did not know HTML before creating this program. We tried to make web pages while learning the use of HTML, JavaScript, and CSS programming languages, while refining and improving the original program. In the process, we found that the web version of the application had many flaws compared to the original Python UI. We spent a lot of time fixing these problems and trying to make the web version as effective as possible. One of the most difficult parts was to make a web version of the stick figure and its joints, which requires javaScript. Since the search program refers to the posture of the stick figure, different UI models cannot be exactly the same. In the web version, we spent more time studying how to obtain the correct position of the joints to improve the search accuracy.

3. SOLUTION

For our project, we have a website in which users can move different parts of the body of a stick figure and get 3 images from our database that are more similar to the stick figure pose. (see Figure 1) On the website, we use HTML, CSS, Javascript and on our server use Python. The server analyzes the stick figure pose by calculating the angles of different parts of the body and compares the angles with the existing images in the database and returns the 3 most similar images.

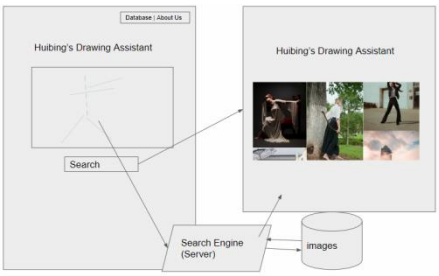

Figure 1. Overview of the solution

Figure 2 shows the overview of our website. We have 5 different website pages but the most important pages are the pictures, about us and and search. The pictures page contains all the images that are stored in our database. The about us page contains the information of our project and the founder. The search page contains the stick figure and the images that are similar to the stick figure. In this search page, users can move the diverse joins on the stick figure to get the desirable images.

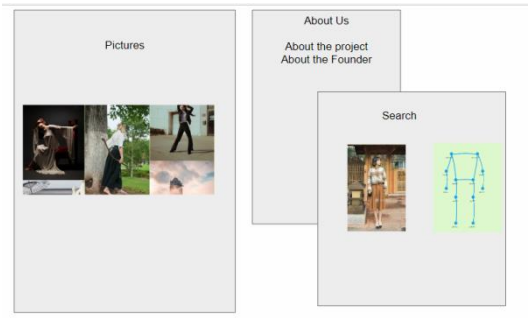

Figure 2. Overview of Website

The backend consists of a database and a server engine. Our database includes about 130 images with different body poses. In order to analyze the images, we use Mediapipe Pose to extract the landmarks of the body. “MediaPipe Pose is a ML solution for high-fidelity body pose tracking, inferring 33 3D landmarks and background segmentation masks on the whole body from RGB video frames utilizing our BlazePose research that also powers the ML Kit Pose Detection API.” [1]. The landmarks consist of the most important part of the body from the eyes to the toes but we focus on shoulders, elbows, wrists, hips, knees and ankles (see Figure 3). After we extract the landmarks from different images, we calculate the angle between diverse joints (see Figure 4) For example, if we calculate the angle between the left shoulder and the left elbow, we use the x and y coordinates of these 2 body parts and use the arctangent to find the angle. Finally, we save all the joint angles of different images in a json file to enhance the performance of the system, so we do not need to consume extra time to fetch all landmarks of each image and calculate the joint angles.

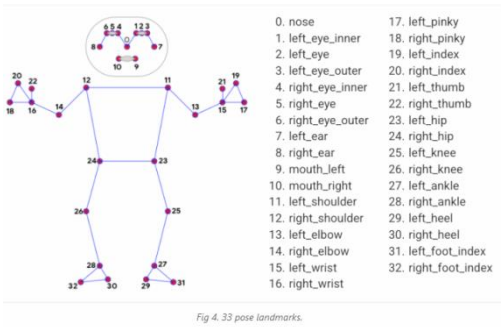

Figure 3. 33 Pose landmarks

```
def mediapipe():
    global imagepath, pictureList
    # with open("data.json","r+") as file:
    #     file.truncate(0)
    posedata = {}
    for picture in pictureList: #this block gets angles for each pic and appends it to dictionary
        posedata[picture] = []
        try:
            for joint1,joint2 in mpPose(f"static/pictures/{picture}"):
                posedata[picture].append(angle(joint1,joint2))
            print(posedata)
        except UnboundLocalError:
            print(picture)
            pass

        json_object = json.dumps(posedata, indent=2) #this block adds a dictionary for each pic to our json file
        with open("data.json", "w") as outfile:
            outfile.write(json_object)
```

Figure 4. Calculate and Save Joint Angles

In order to search for the three images that are more similar to the stick figure, we calculate the joint angles of the stick figure and fetch the angles of all images that are saved in the json file. (see Figure 5) Then we assign the weight of the joint angles based on the most important body joints which are the angles between left shoulder and left elbow, left elbow and left wrist , right shoulder and right elbow, right elbow and right wrist, left hip and left knee, and right hip and right knee. Finally, we sum all the weights of different images and select the three weights that have minimal values.

```
def search(joints):
    f = open('data.json')
    data = json.load(f)
    for key in data:
        for i,val in enumerate(data[key]):
            if abs(val) > 1.3:
                data[key][i] = abs(val)

    right_shoulder = Joint(joints['right_shoulder']['x'],joints['right_shoulder']['y'])
    left_shoulder = Joint(joints['left_shoulder']['x'],joints['left_shoulder']['y'])
    right_waist = Joint(joints['right_waist']['x'],joints['right_waist']['y'])
    left_waist = Joint(joints['left_waist']['x'],joints['left_waist']['y'])
    right_elbow = Joint(joints['right_elbow']['x'],joints['right_elbow']['y'])
    left_elbow = Joint(joints['left_elbow']['x'],joints['left_elbow']['y'])
    right_wrist = Joint(joints['right_wrist']['x'],joints['right_wrist']['y'])
    left_wrist = Joint(joints['left_wrist']['x'],joints['left_wrist']['y'])
    right_knee = Joint(joints['right_knee']['x'],joints['right_knee']['y'])
    left_knee = Joint(joints['left_knee']['x'],joints['left_knee']['y'])
    right_ankle = Joint(joints['right_ankle']['x'],joints['right_ankle']['y'])
    left_ankle = Joint(joints['left_ankle']['x'],joints['left_ankle']['y'])

    for key in data:
        if len(data[key]) > 0:
            differences[key] = []
            for i in range(len(data[key])):
                if i in [1,2,3,4,8,10]:
                    differences[key].append(abs(Angles[i]-data[key][i])*2)
                else:
                    differences[key].append(abs(Angles[i] - data[key][i]))

    for key in differences:
        sum_of_diff[key] = 0
        for diff in differences[key]:
            sum_of_diff[key] += diff
    result = min(sum_of_diff, key=sum_of_diff.get)
```

Figure 5. Search images

To develop our website, we use HTML, CSS and Javascript which consist of 5 different web pages. Home page is the landing page that contains a button to redirect the webpage to the search page. The about page contains the information of the founder and the application. The picture page contains all the images that are saved in the database. Search page is where users manipulate the joints of the stick figure to get the desirable images.

In order to interact in the search web page, users would move different parts of the body, right/left shoulder, right/left elbow, right/left wrist, right/left hip, right/left knee and right/left ankle. After they finished moving all body parts, they would click on the search button to send the information to the server and receive the images that are more similar to the stick figure. The users can see 3 different images by clicking the next/previous buttons. (See Figure 6)

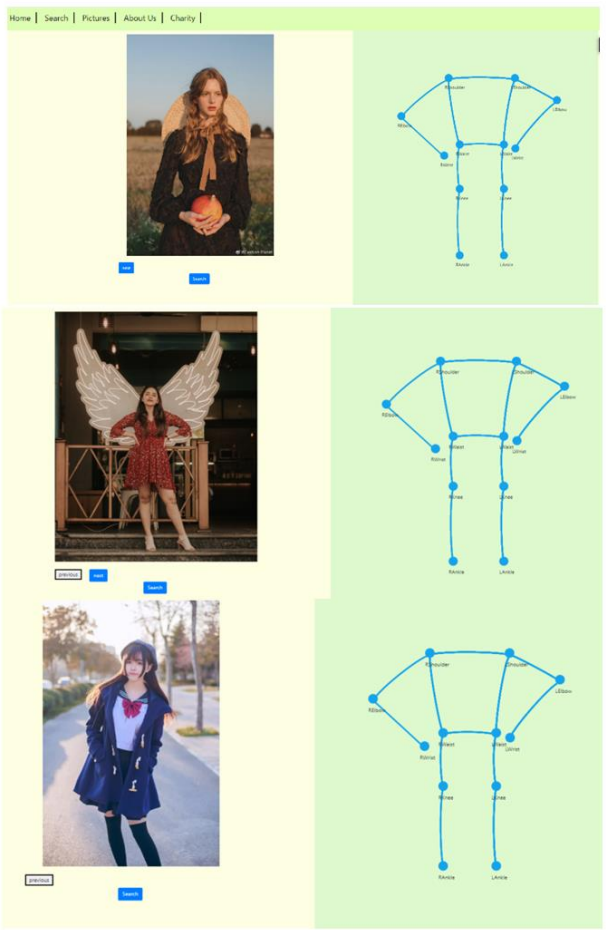

Figure 6. Resulted images from stick figure

In order to draw the stick figure, we defined the joints with a default size and location. To connect the different joints, we used bezier curve elements. To track a joint, we observe the mouse event, so when users click hold a joint and move the mouse, we can get the position of the mouse on the screen and translate and move the connector and the joint to the corresponding position.

```
    function onMouseMove(e) {
        e.preventDefault();
        if (the_moving_div == "") return;
        var d = document.getElementById(the_moving_div);
        d.style.left = d.offsetLeft + e.clientX - the_last_mouse_position.x + "px";     // move the div by however much the mouse moved
        d.style.top  = d.offsetTop  + e.clientY - the_last_mouse_position.y + "px";
        the_last_mouse_position.x = e.clientX;                                          // remember where the mouse is now
        the_last_mouse_position.y = e.clientY;
        joints[the_moving_div].x = d.offsetLeft + e.clientX - the_last_mouse_position.x - WIDTH
        joints[the_moving_div].y = d.offsetTop + e.clientY - the_last_mouse_position.y - HEIGHT
        d = document.getElementById(the_moving_div);
        setJoint(d, joints[the_moving_div].x , joints[the_moving_div].y)
        drawJoints();
        console.log(joints[the_moving_div])
    }

function drawConnector(ctx,canvas,joint1, joint2) {
    height = 55;
    ctx.moveTo(joint1.offsetLeft - window.innerWidth/2+ joint1.clientWidth/2, joint1.offsetTop-  height + joint1.clientHeight/2);
    ctx.bezierCurveTo(joint1.offsetLeft- window.innerWidth/2, joint1.offsetTop-  height,
                      joint2.offsetLeft- window.innerWidth/2, joint2.offsetTop-  height,
                      joint2.offsetLeft - window.innerWidth/2+ joint2.clientWidth/2, joint2.offsetTop-  height + joint2.clientHeight/2);
    ctx.stroke();
}
```

Figure 7. Draw and move connectors and joints

4. EXPERIMENT

4.1. Experiment 1

In our research, we performed 2 different experiments. In the first experiment, we developed 2 algorithms and compared how accurate both algorithms are when they retrieve the images after receiving the joints of the stick figure. The second experiment, we performed a data analysis of user satisfaction to get some feedback from the app.

In this experiment, we used a local database to store around 130 images of people with different poses and used about 15 stick figure poses to get the match images. We used Python and Mediapipe to write the 2 image matching algorithms. The algorithms that we used are the k-mean clustering and the sum of different angles. For the images stored in the local database and the stick figure, we extracted the landmarks and calculated the angles. Then fitted both models with the angle information of the images and the stick figure.

After we tested both algorithms, we observed that the sum of difference algorithm was more accurate than the k-means clustering with 87% and 40%, respectively. (See Fig 8) We believe that adding different weights to each body part can improve the result of the images.

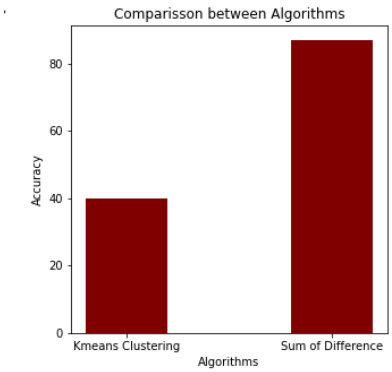

Figure 8. Comparison between Algorithm

4.2. Experiment 2

For this experiment, we did a data analysis to obtain feedback from the users. We use 10 High School students who are interested in drawing and painting to try our application After they use our tool, we provided a questionnaire asking them about:

1. How was the experience using the interface?. (Easy, neutral or complicated)
2. Does the application provide the desirable images? (1-5)

Fig 9 shows the result for the interface feedback. When we asked the user about the user interface and rated our application, we obtained that six High School students responded that the interface was easy to use, three of them responded that it was not too easy or complicated and one student responded that it was complicated.

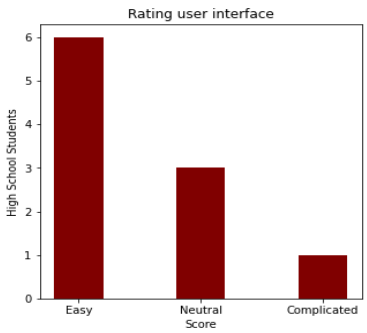

Figure 9. Rating user Interface

In the feedback, we also asked the students to rate our application to observe if the app provides the desirable images or not. They can choose between 1 to 5. 1 means that it doesn't fulfill his/her expectation, 5 means that the app provides his/her desirable images and between 1 to 5 means that it provides some of the images. The result shows us that three students rated 1, three students rated 3 and four rated 5.

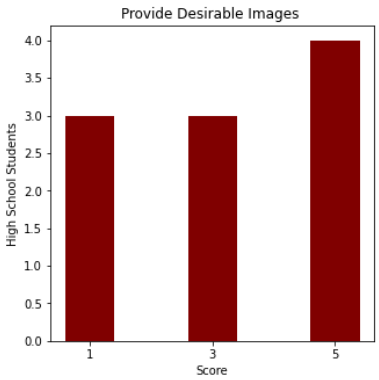

Figure 10. Provide Desirable Image

When we tested different algorithms for matching the human pose from an image and the stick figure image, experiment 1 shows us that the sum of different algorithms performs better than the K-means clustering. We believe that both algorithms show these results because having different weights of the angles for different images can help us to focus more on specific joints of the body and obtain the desirable images in comparison to the k-means that all the joints have the same weight.

For experiment 2, we want to observe if the application fulfills the objective of the project, which is to provide an easy tool in which users can search an image with a specific pose. The result shows us that most of the users believe that the tool is easy to use. Also, most of the users feel that the tool retrieves the desirable images with a specific pose. Thus, our application reaches our goal to provide a tool that can retrieve human pose images.

5. RELATED WORK

Hayashi et al. performed research about the pose estimate in sport activities [2]. In their experiment, they focused on Hockey and American Football videos to predict the pose estimate of the upper part of the bodies. The result shows that the pose estimate is accurate using the random decision forest classifiers. In my research, we focus on the pose estimate of images instead of videos to extract the

pose estimate from the stick figure and static images. Also, Hayashi et al. focus only on the upper body parts while our research focused on the whole body parts.

Borkar et al. proposed a system that can extract the pose estimate of people images practicing dance or yoga and compare it with people's realtime pose by using a webcam [3]. For the comparison of the yoga or dance images, they use PoseNet to extract the pose estimate and calculate the angles of different body parts . They use shoulder coordinates and the wrists coordinates to locate the arms, hips and knees coordinates, and hips and the ankles coordinates for the legs. In our research, we use a similar approach. We calculate the angles of the arms and legs; however, we use different parts of the body to calculate the angle of the arms. We use the angle between the shoulders and elbows and the elbows to wrist, which gives us higher accuracy of the arms location.

Pauzi et al. present a method to detect the movement for body injury during heavy workload [4]. They analyze real time video and use Mediapipe to extract the joint of the body and calculate velocity and the angle of joints which is important to know if a person has some injuries or not. In our study, we use Mediapipe to extract joints of the body. As different from Pauzi et al. paper, we are not focused on real time video. We compare static images and stick figure poses to match and return the images that are more similar.

6. CONCLUSIONS

Nowadays, drawing has become more popular and many people practice this activity. Thus, there are a lot of software and tools that they can use. However, there are many painters that have encountered a lot of challenges while they need to draw the human body with a specific pose. We proposed a tool in which people can search images of the human body with a specific pose. In this tool, people can move any part of the human joints from a stick figure to obtain the desirable image which pose is similar to the stick figure. We use Mediapipe to extract the human body and stick figure landmarks and compare them by using the sum of difference algorithm [14].

We performed two different experiments. The first experiment, we investigated the performance and accuracy of 2 different algorithms to match the stick figure pose and the human pose from diverse images. The experiment shows that the sum difference algorithm performs better than the K-mean Clustering algorithm with 87% and 40% [15]. The other experiment, 10 High school students provided feedback after they had used the tool. With this data, we performed a data analysis to observe the user acceptance of the tool. The result shows that six students believe that the tool was easy to use in terms of user interface. Also, five students believe that the tool provides the desirable images.

However, we encountered some limitations in terms of accuracy. Our sum of difference model accuracy was about 87%, so some students were not completely satisfied with our tool. Some of the reason is that the dataset contains about 130 images which are not enough to retrieve the specific matching human pose images. Also, we have some limitations in the number of participants to try our tool. We did the experiment during the summer, so we could not advertise our tool in some high schools, so we could not obtain more feedback about our tool.

For our future work, we would like to increase the number of questions in our questionnaire, and advertise our tool in diverse high schools during the school season, so we can obtain more feedback about our tool. Also, we would like to include more images to make our sum of difference model more accurate. Finally, we would like to optimize our K-means clustering algorithm, so the model can be more accurate, and do some more experiments with different machine learning models to optimize our engine.

AUTHOR

HuiBingXie is a junior student studying in Northwood High School.

REFERENCES

[1] Singh, Amritanshu Kumar, Vedant Arvind Kumbhare, and K. Arthi. "Real-Time Human Pose Detection and Recognition Using MediaPipe." International Conference on Soft Computing and Signal Processing. Springer, Singapore, 2021.

[2] Hayashi, Masaki, et al. "Head and upper body pose estimation in team sport videos." 2013 2nd IAPR Asian Conference on Pattern Recognition. IEEE, 2013.

[3] Borkar, PradnyaKrishnanath, Marilyn Mathew Pulinthitha, and A. Pansare. "Match Pose-A System for Comparing Poses." International Journal of Engineering Research and Technology (IJERT) 8.10 (2019).

[4] Pauzi, AinunSyarafana Binti, et al. "Movement Estimation Using MediapipeBlazePose." International Visual Informatics Conference. Springer, Cham, 2021.

[5] Edwards, Betty. "Drawing on the Right Side of the Brain." CHI'97 Extended Abstracts on Human Factors in Computing Systems. 1997. 188-189.

[6] Jiang, Min, and Guodong Guo. "Body weight analysis from human body images." IEEE Transactions on Information Forensics and Security 14.10 (2019): 2676-2688.

[7] Paul, Christiane. "New media in the white cube and beyond: Curatorial models for digital art." Leonardo Reviews Quarterly 1.2010 (2008): 33.

[8] Black, Joanna, and Kathy Browning. "Creativity in digital art education teaching practices." Art Education 64.5 (2011): 19-34.

[9] Hanson, Jill M., et al. "Fast dynamic imaging using two reference images." Magnetic resonance in medicine 36.1 (1996): 172-175.

[10] Kakadiaris, Ioannis A., and Dimitri Metaxas. "Three-dimensional human body model acquisition from multiple views." International Journal of Computer Vision 30.3 (1998): 191-218.

[11] Wang, Liang, and David Suter. "Analyzing human movements from silhouettes using manifold learning." 2006 IEEE International Conference on Video and Signal Based Surveillance. IEEE, 2006.

[12] Tkachenko, Volodymyr, Aleksandra Kuzior, and AleksyKwilinski. "Introduction of artificial intelligence tools into the training methods of entrepreneurship activities." Journal of Entrepreneurship Education 22.6 (2019): 1-10.

[13] Lugaresi, Camillo, et al. "Mediapipe: A framework for building perception pipelines." arXiv preprint arXiv:1906.08172 (2019).

[14] Lugaresi, Camillo, et al. "Mediapipe: A framework for perceiving and processing reality." Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR). Vol. 2019. 2019.

[15] Ahmad, Amir, and Lipika Dey. "A k-mean clustering algorithm for mixed numeric and categorical data." Data & Knowledge Engineering 63.2 (2007): 503-527.