Methodology for teaching development of web-based augmented reality with integrated machine learning models

Serhiy O. Semerikov[1,2,3,4,5], Mykhailo V. Foki[1], Dmytro S. Shepiliev[1], Mykhailo M. Mintii[6], Iryna S. Mintii[7,2,1,8,6,3,5] and Olena H. Kuzminska[9,1]

[1] *Kryvyi Rih State Pedagogical University, 54 Universytetskyi Ave., Kryvyi Rih, 50086, Ukraine*
[2] *Institute for Digitalisation of Education of the NAES of Ukraine, 9 M. Berlynskoho Str., Kyiv, 04060, Ukraine*
[3] *Zhytomyr Polytechnic State University, 103 Chudnivsyka Str., Zhytomyr, 10005, Ukraine*
[4] *Kryvyi Rih National University, 11 Vitalii Matusevych Str., Kryvyi Rih, 50027, Ukraine*
[5] *Academy of Cognitive and Natural Sciences, 54 Universytetskyi Ave., Kryvyi Rih, 50086, Ukraine*
[6] *Kremenchuk Mykhailo Ostrohradskyi National University, 20 University Str., Kremenchuk, 39600, Ukraine*
[7] *University of Łódź, 68 Gabriela Narutowicza Str., 90-136 Łódź, Poland*
[8] *Lviv Polytechnic National University, 12 Stepana Bandery Str., Lviv, 79000, Ukraine*
[9] *National University of Life and Environmental Sciences of Ukraine, 15 Heroiv Oborony Str., Kyiv, 03041, Ukraine*

Abstract

Augmented reality (AR) is an emerging technology with many applications in education. Web-based augmented reality (WebAR) provides a cross-platform approach to deliver immersive learning experiences on mobile devices. Integrating machine learning models into WebAR applications can enable advanced interactive effects by responding to user actions. However, little research exists on effective methodologies to teach students WebAR development with integrated machine learning. This paper proposes a methodology with three main steps: (1) Integrating standard TensorFlow.js models like handpose into WebAR scenes for gestures and interactions; (2) Developing custom image classification models with Teachable Machine and exporting to TensorFlow.js; (3) Modifying WebAR applications to load and use exported custom models, displaying model outputs as augmented reality content. The methodology is designed to incrementally introduce machine learning integration, build an understanding of model training and usage, and spark ideas for using machine learning to augment educational content. The methodology provides a starting point for further research into pedagogical frameworks, assessments, and empirical studies on teaching WebAR development with embedded intelligence.

Keywords

web-based augmented reality, WebAR, machine learning, TensorFlow.js, Teachable Machine, educational technology

1. Introduction

Web-based Augmented Reality (WebAR) is one of the most common ways to combine the real and the virtual on mobile Internet devices [1, 2]. The development of web-based augmented reality applications differs from other development methods in that it is cross-platform and does not require the installation of developed applications, which significantly increases the level of software mobility compared to traditional mobile applications [3, 4].

Currently, the world's most famous non-profit library for WebAR development is AR.js [5], founded by Jerome Etienne (for example, [6] provides a systematic description of the possibilities of using AR.js for the development of professional competences of future teachers of STEM disciplines), but HiuKim

CoSinE 2024: 11th Illia O. Teplytskyi Workshop on Computer Simulation in Education, co-located with the XVI International Conference on Mathematics, Science and Technology Education (ICon-MaSTEd 2024), May 15, 2024, Kryvyi Rih, Ukraine

semerikov@gmail.com (S. O. Semerikov); ierehon575@gmail.com (M. V. Foki); sepilevdmitrij@gmail.com (D. S. Shepiliev); mykhailo.mintii@gmail.com (M. M. Mintii); mintii@acnsci.org (I. S. Mintii); o.kuzminska@nubip.edu.ua (O. H. Kuzminska)
https://acnsci.org/semerikov (S. O. Semerikov); http://ivm.krnu.org/mintij-mm-prof-dosygnennya (M. M. Mintii); https://acnsci.org/mintii/ (I. S. Mintii); https://nubip.edu.ua/node/73749 (O. H. Kuzminska)
0000-0003-0789-0272 (S. O. Semerikov); 0000-0001-6913-8073 (D. S. Shepiliev); 0000-0002-0488-5569 (M. M. Mintii); 0000-0003-3586-4311 (I. S. Mintii); 0000-0002-8849-9648 (O. H. Kuzminska)

Yuen [7], one of the developers of AR.js, created a new library called MindAR [8], which is more compact and technologically advanced, but, unlike AR.js, is little known.

AR.js and MindAR are built on the classic ARToolKit and OpenCV engines, respectively, which are currently the industry standard. At the same time, while AR.js is focused on processing primarily simple markers up to 16×16, MindAR is focused on natural images of complex structures. Another feature of MindAR that makes it an appropriate learning tool is the inclusion of the well-known TensorFlow machine learning library [9], which provides potential opportunities for integrating machine learning models into WebAR applications to create highly interactive and exciting effects, such as using hand gestures or facial expressions to control AR content.

The aim of the study is to develop the methodology for teaching the development of augmented reality for the Web with integrated machine learning models.

The main **objectives of the study** are as follows:

1. Perform a bibliometric analysis of sources from educational applications of WebAR.
2. Choose tools for developing augmented reality for the Web.
3. Develop and test a methodology for developing WebAR applications for face tracking.
4. Develop and test a methodology for integrating machine learning models into WebAR applications.

2. Bibliometric analysis of sources from educational applications of WebAR

To perform a systematic bibliometric analysis for the queries “WebAR” and “Web-based augmented reality for education”, VOSviewer version 1.6.18 [10] was used.

As a data source for the first query, Crossref was selected with a search by document titles, which made it possible to select 19 documents from 2017-2022 (date of request: 26.11.2022). The selected documents were analysed by the times they were co-cited with other documents.

Out of 92 sources cited in 19 documents, 26 are cited together more than once, forming only 1 cluster (figure 1), which includes works [1, 2, 11], performed under the supervision of Serhiy O. Semerikov.

Scopus was chosen as the data source for the second query, with a search by titles, abstracts and keywords, which made it possible to select 93 documents from 2001-2023 (figure 2), 66 of which are from the last five years. The majority of them are journal articles (58 [12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69]), the smaller part is books (4 [70, 71, 72, 73]) and articles in conference proceedings (31 [74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104]).

Out of 301 authors of 93 documents, 27 were cited twice or more times and 9 were cited three or more times. figure 3 shows the semantic network of keywords in the documents for the query “Web-based augmented reality for education”. The distribution of keywords by clusters (figure 4) is shown in table 1.

The first cluster (highlighted in red in figure 4 and table 1) connects the **basic concepts of augmented reality in education**: augmented and virtual reality with education (including medical education) and human learning, including the use of smartphones.

Augmented reality is a systemic element – it connects all the clusters and is itself connected to all other concepts.

In the analysed documents, *virtual reality* is not linked to traditional, mobile, and Internet/web-based learning. It is essential to distinguish virtual reality from virtual learning environments, which include these concepts.

The concept of *education* is also almost universal – it is not only associated with user interfaces and AR applications.

The links of *medical education* with other clusters are quite revealing: in the second cluster – with the concepts of curricula, computer-aided instruction, education computing, e-learning and students; in the third – with websites and pedagogical augmented reality technology, in the fourth – with distance education.

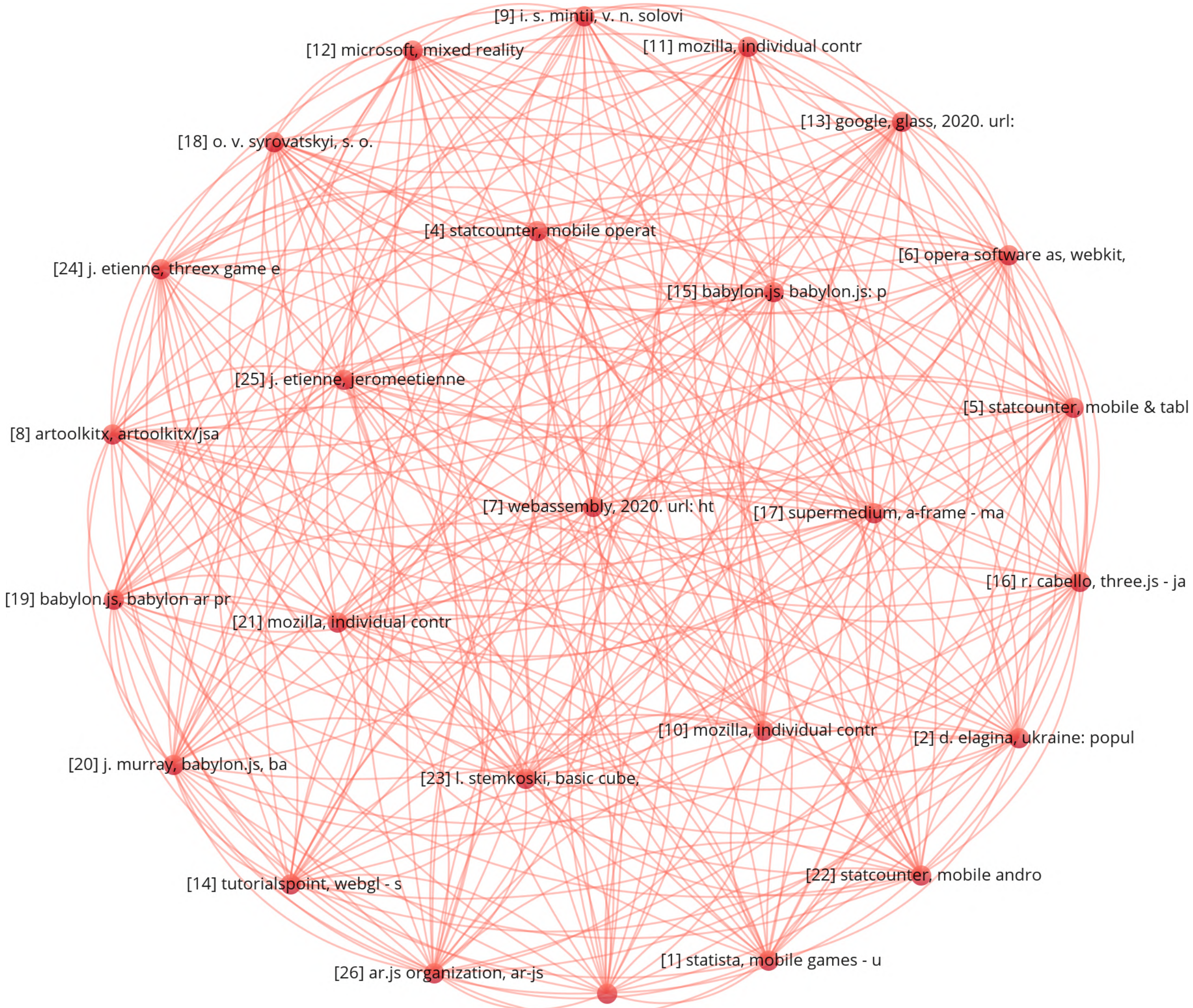

[9] i. s. mintii, v. n. solovi
[12] microsoft, mixed reality
[11] mozilla, individual contr
[13] google, glass, 2020. url:
[18] o. v. syrovatskyi, s. o.
[4] statcounter, mobile operat
[6] opera software as, webkit,
[24] j. etienne, threex game e
[15] babylon.js, babylon.js: p
[25] j. etienne, jeromeetienne
[5] statcounter, mobile & tabl
[8] artoolkitx, artoolkitx/jsa
[7] webassembly, 2020. url: ht
[17] supermedium, a-frame - ma
[16] r. cabello, three.js - ja
[19] babylon.js, babylon ar pr
[21] mozilla, individual contr
[10] mozilla, individual contr
[2] d. elagina, ukraine: popul
[20] j. murray, babylon.js, ba
[23] l. stemkoski, basic cube,
[22] statcounter, mobile andro
[14] tutorialspoint, webgl - s
[1] statista, mobile games - u
[26] ar.js organization, ar-js

Figure 1: The semantic network of links in the documents for the query “WebAR”.

Learning (in the sense of studying) is related in the second cluster to teaching, students, education computing, computer-aided instruction, e-learning and user interfaces, and in the third cluster to websites and motivation. This concept has no direct links to distance education.

The concept of a *human(s)* (person(s)) outside their cluster is linked to students and e-learning in the second cluster and websites in the third.

Outside of its cluster, *Internet/web-based learning* is only associated with traditional teaching in the second cluster.

Finally, *smartphones* are linked in the second cluster to teaching, students, education computing, e-learning and engineering education, and in the third cluster to websites and augmented reality applications.

The second cluster (highlighted in green in figure 4 and table 1) connects the **concepts of learning environment design**: teaching, engineering education, computer-aided instruction, e-learning, students, mobile learning, learning environments, education computing, and curricula.

Central to the second cluster are the concepts of *“e-learning”* and *“students”*, which are also almost universal – formally, they are not associated only with Internet/web-based learning due to their synonymity with e-learning.

Computer-aided instruction is related to the concepts of the first (augmented and virtual reality, education (including medical) and learning) and third (motivation, websites, learning systems, interactive learning environments, augmented reality applications, augmented reality technology) clusters.

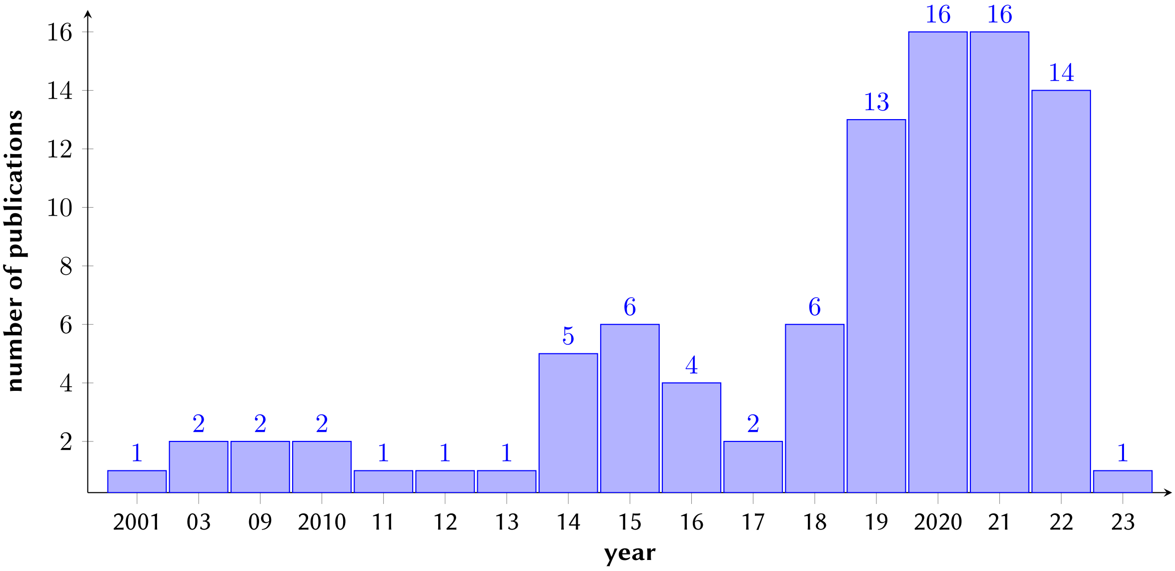


Figure 2: Distribution of documents by year (query “Web-based augmented reality for education”).

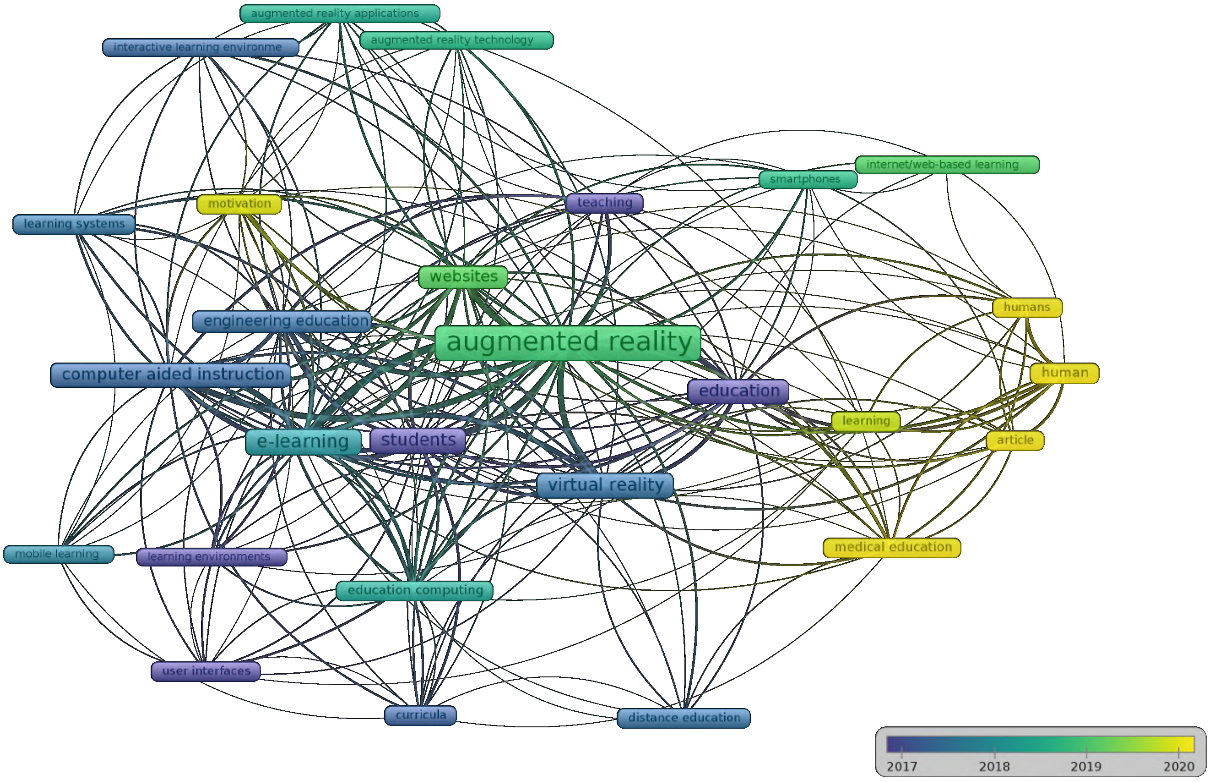


Figure 3: The semantic network of keywords in documents by the query “Web-based augmented reality for education”.

The concept of *teaching* is linked in the first cluster to augmented reality, education and learning, smartphones and Internet/web-based learning, and in the third cluster to websites, augmented reality applications and augmented reality technology.

Engineering education is related in the first cluster to augmented and virtual reality, education and smartphones, and all concepts of the third and fourth clusters.

Education computing is related in the first cluster to augmented and virtual reality, education (including medical) and learning, smartphones, and in the third cluster to motivation, learning systems and websites,

Table 1
Distribution of keywords by clusters (documents by query “Web-based augmented reality for education”).

Cluster 1	Cluster 2
article	computer-aided instruction
augmented reality	curricula
education	e-learning
human	education computing
humans	engineering education
internet/web-based learning	learning environments
learning	mobile learning
medical education	students
smartphones	teaching
virtual reality	user interfaces

Cluster 3	Cluster 4
augmented reality applications	distance education
augmented reality technology	
interactive learning environments	
learning systems	
motivation	
websites	

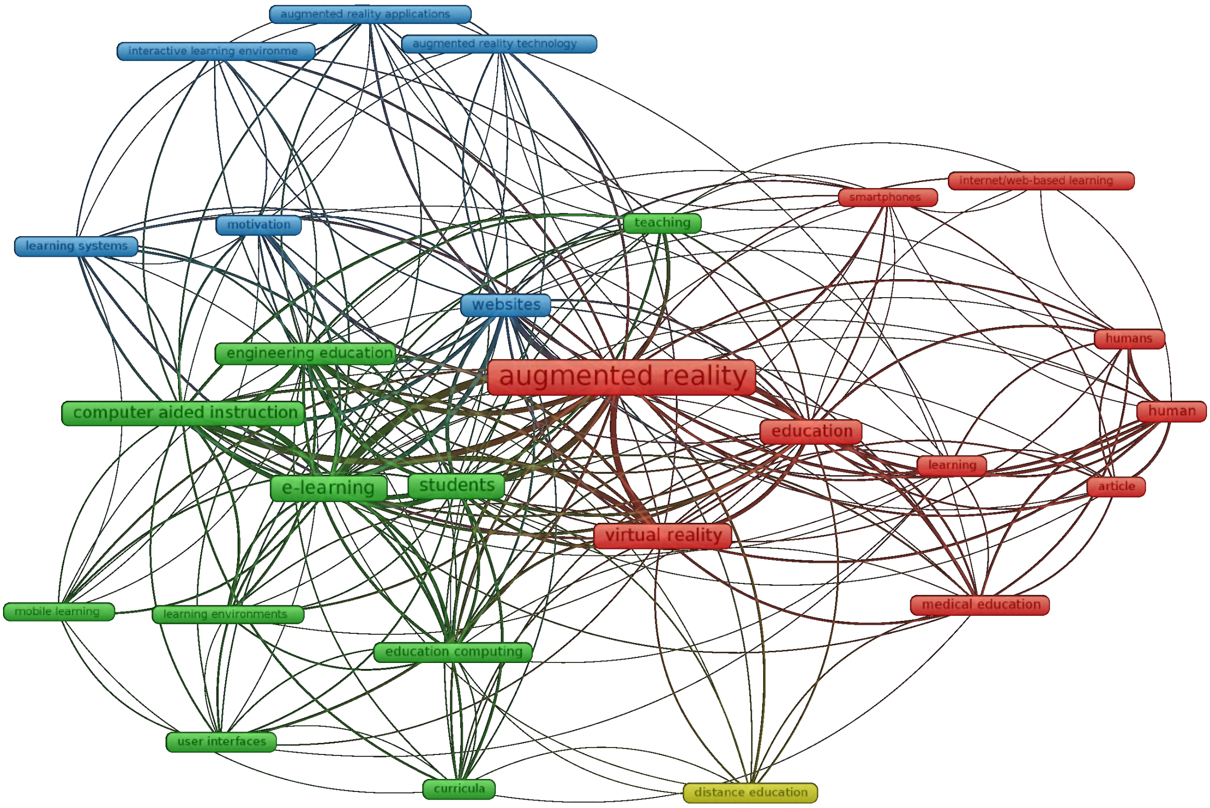


Figure 4: Distribution of keywords by cluster.

and the fourth – with distance education.

Outside their cluster, *learning environments* are only related to augmented and virtual reality education from the first cluster and websites from the third.

Similarly, *mobile learning* is related to education and augmented reality in the first cluster and motivation, websites and learning systems in the third.

User interfaces have links to the concepts of the first (learning, augmented and virtual reality) and third (motivation, websites) clusters.

The *Curricula* are related to education (including medical education), augmented and virtual reality in the first cluster, websites in the third cluster, and distance education in the fourth cluster.

The third cluster (highlighted in blue in figure 4 and table 1) connects the **concepts of immersive learning environment implementation**: websites, motivation, learning systems, interactive learning environments, augmented reality applications and augmented reality technology.

Central to the third cluster are *websites*, which are almost universal concepts – formally, they are not associated only with Internet/web-based learning due to the overlap of the relevant concepts.

The concept of *motivation* is related in the first cluster to augmented and virtual reality, education and learning, and in the third cluster to e-learning and mobile learning, education computing, user interfaces, computer-aided instruction, students and engineering education.

Learning systems are related in the first cluster to augmented and virtual reality and education and in the third cluster to e-learning and mobile learning, education computing, computer-aided instruction, student-centred teaching and engineering education.

Interactive learning environments also have similar links: in the first cluster, with augmented and virtual reality and education, and in the third cluster, with e-learning, computer-aided instruction, students and engineering education.

Naturally, *augmented reality applications* are related to augmented reality and smartphones in the first cluster and to e-learning, computer-aided instruction, teaching, students and engineering education in the second.

Augmented reality technology are related in the first cluster to augmented and virtual reality and education (including medical education) and in the second cluster to e-learning, computer-aided instruction, teaching, students and engineering education.

The fourth cluster (highlighted in yellow in figure 4 and table 1) contains the **concept of distance education**, which is linked in the first cluster to the concepts of augmented and virtual reality and the concept of education (including medical education), in the second cluster to the concepts of student, engineering education, education computing, e-learning and curricula, and in the third cluster to the concept of website.

The analysis of the distribution of concepts by the density of links (figure 5) and time makes it possible to determine that the oldest (before 2015) studies focused on user interfaces and their application in education. In 2016, the focus shifted to studying the impact of teaching in learning environments on students. In 2017, the research actualised the concepts of virtual reality, interactive learning environments, curricula, and computer-aided instruction in engineering education. The focus of research in 2018 was on education computing, the use of smartphones, augmented reality applications and pedagogical augmented reality technology.

WebAR is the focus of research in 2019, with studies addressing the use of smartphones, online/web-based learning and augmented reality. In 2020, the impact of the COVID-19 pandemic added to the issues of learning motivation and medical education. A new element of recent research is human augmentation.

3. Augmented reality development tools for the Web

3.1. Setting up a web server and remote debugger

The main development tools to develop in HTML and JavaScript are a simple text editor and a web browser, where you can open a regular HTML web page saved locally.

However, this may not work for applications that require a camera. In addition, you may want to test applications on your own mobile devices from time to time, so it is best to install a local web server like Simple Web Server.

It may be helpful to select the HTTPS protocol in the advanced settings – without it, the mobile device may not be able to access the camera.

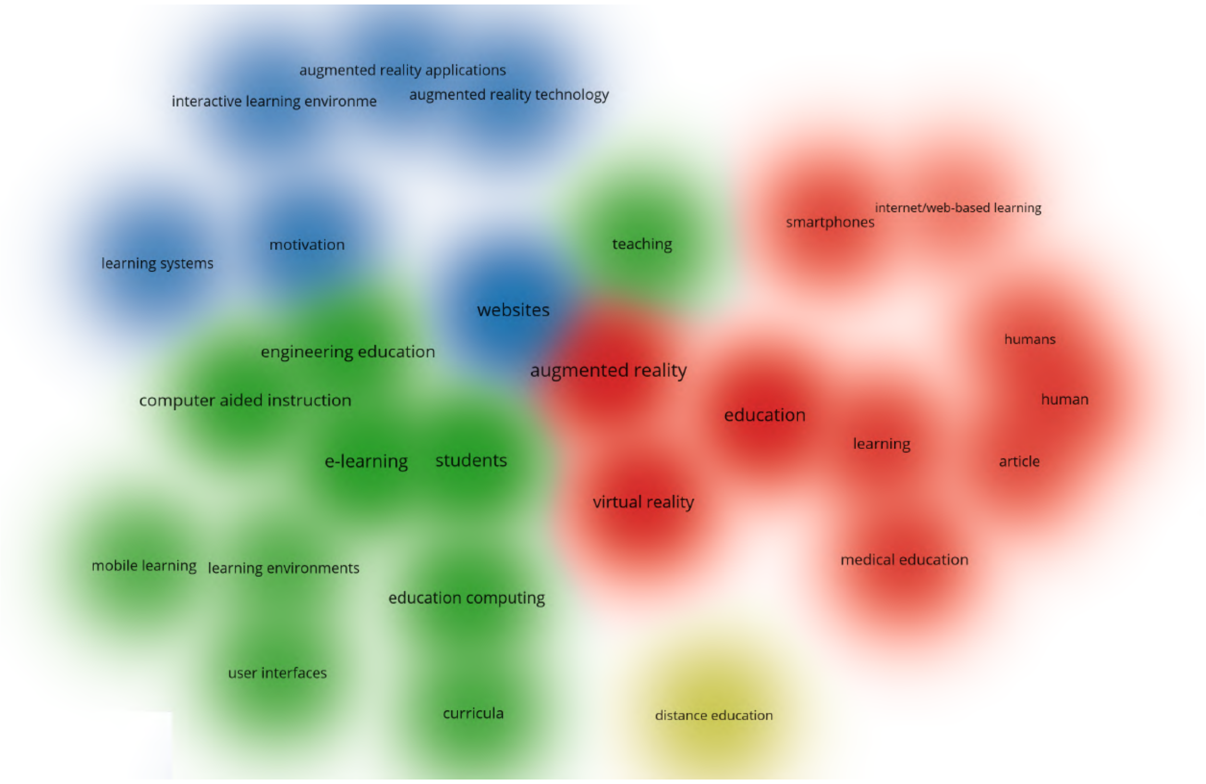


Figure 5: The density of keyword connections for the query "Web-based augmented reality for education".

Doing all the development and testing work directly on a desktop browser is possible, but sometimes, it is still worth trying on a mobile phone.

If the devices are connected to the same local area network that does not have a firewall, there is no problem accessing the web server. However, if the network access point is behind a firewall, you can use ngrok to redirect traffic from the restricted port.

After installing ngrok and creating an account on the website [105], you need to register the ngrok agent [106] and start it, specifying the protocol (e.g. HTTP) and the port number that the firewall denies access to (e.g. 8887).

Once started, ngrok provides a global HTTPS Internet link – but only while the local web server and the ngrok redirect are running.

Traditionally, debugging web applications involves viewing the web browser console, which displays notifications related to debugging the application.

However, it may be challenging on a mobile device. Here, RemoteJS [107] will help – by clicking the `Start Debugging` button after going to the site, we will get the RemoteJS agent code like this:

```
<script data-consolejs-channel="9817ec3e-a3f7-fbe3-3836-e2e2d07d5c99"
  src="https://remotejs.com/agent/agent.js">
</script>
```

This code should be copied and pasted directly into the web page.

After that, all debug messages will be sent to the web page at https://remotejs.com/viewer/agent_code, where `agent_code` is the value of the `data-consolejs-channel` variable.

3.2. Application of a graphical library for augmented reality on the Web

WebGL [108] is a JavaScript API for rendering 3D graphics in browsers. It is a cross-platform display standard supported by all major browsers. However, low-level WebGL code is difficult to read and write, so more user-friendly libraries have been created.

Three.js [109] is one such library. Its author, Ricardo Miguel Cabello, also known as mrdoob, is one of the pioneers of WebGL, so this library is often used when building other libraries. Most WebAR SDKs support Three.js, so it is a must-have language for effectively developing augmented reality web applications.

To understand how Three.js works at a high level, it is useful to draw an analogy with the work of a photographer or film director who:

1) customises the scene by placing objects on it;
2) moves the camera to capture footage from different positions and angles.

Three.js is not a specialised library for augmented reality – it contains much more functionality, including that which is more suitable for web VR (lighting, cameras, etc.) (figure 6).

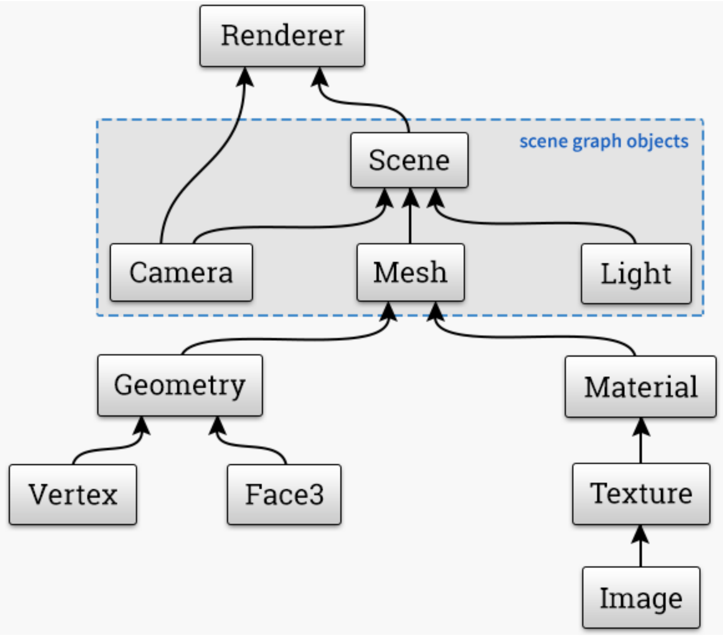


Figure 6: The general structure of Three.js.

As shown in figure 6, the basis is a scene where objects are created in three steps:

1) determination of object geometry – position vectors, colours, etc.: e.g., `BoxGeometry` is responsible for the rectangular parallelepiped;
2) definition of the material – the way the object is rendered (its optical properties – colour, texture, gloss, etc.): for example, `MeshBasicMaterial` corresponds to a material that has its colour and does not reflect rays;
3) geometry and material composition is performed using `Mesh`.

The renderer will display the 3D model on the canvas, considering the material, texture and lighting. For WebAR applications to work, the scene needs to be transparent so that the video stream from the camera can be overlaid. This is achieved by setting the `alpha` parameter to `true` in the `WebGLRenderer` class constructor.

Rendering itself is performed by the `render` method, which displays the projection of the scene onto the canvas (`canvas` element) from the camera's point of view.

Connect the video stream before linking a canvas to an HTML page for WebAR applications.

figure 7 shows the first implementation of WebAR, in which a real object from the camera is supplemented with a virtual object.

Placing a canvas over the video is the basis of WebAR. The only thing that needs to be added is displaying the object in a more appropriate location and updating its position according to the camera signal, i.e. object tracking.

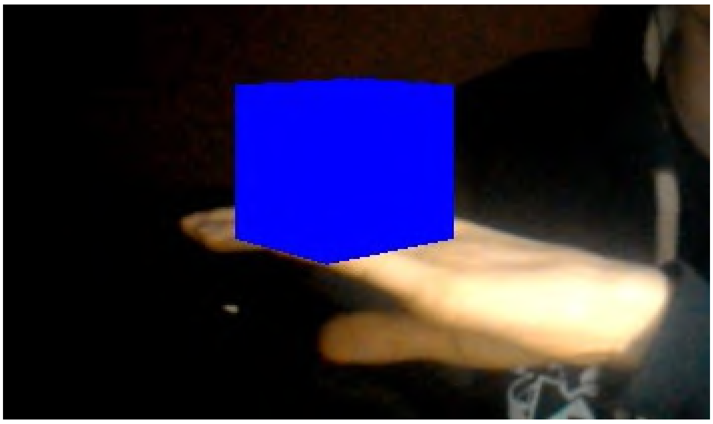


Figure 7: Video stream and model overlay.

3.3. Setting up a library for augmented reality on the Web

You can change the position of the image by moving the virtual camera, changing its position (coordinates), and tilting it. Appropriate changes require tracking objects, so it is expected to classify augmented reality into marker-based, markerless, location-based, etc.

HiuKim Yuen offers a classification of augmented reality by the type of tracking.

The first type is *image tracking*: in this type, virtual objects appear on top of target images, which can be barcode-like, which have a predefined structure, and natural, which can be anything.

The images do not have to be printed or on-screen – there can even be augmented reality T-shirts [110].

The second type of augmented reality is *face tracking*, where objects are attached to the human face. Examples include Instagram filters, Google Meet, social media campaigns, apps for trying on virtual accessories, etc.

The third type of augmented reality is *world tracking*, also known as markerless augmented reality. With this type of tracking, augmented reality objects can be placed anywhere, not limited to a specific image, face, or physical object.

World tracking applications continuously capture and track the environment and estimate the physical position of the application user. Augmented reality objects are often attached to a specific surface, such as the ground.

Location-based augmented reality, known for Pokémon GO, Ingress etc., involves linking content to a specific geographical location – latitude and longitude. Usually, these apps track the environment, as the augmented content is usually attached to the ground, and the location-based part is rather an additional condition that triggers the tracking of the environment (or a face) in a specific location.

Other types of tracking can be defined, such as 3D object tracking, hand tracking, etc.

Despite the variety of libraries for augmented reality, their main task is to determine the position of the virtual camera following the tracked object, as illustrated by the following pseudocode:

```
const ar_engine = new SOME_AR_ENGINE();
while(true) {
  await nextVideoFrameReady();
  const {position, rotation} = ar_engine.computeCameraPose(video);
  camera.position = position;
  camera.rotation = rotation;
}
```

First, you need to initiate a library – a specific AR engine – and get a link to it. Then, in a continuous loop, wait for a frame from the video stream of the real camera, determine its position (tilt coordinates), and move the virtual camera on the canvas to the same position.

Often, however, it is not the virtual camera that moves but the objects on the scene. In this case, the position of the tracked object is determined, rather than the real camera, and then the virtual reality object is moved to the same position as the tracked object:

```
const ar_engine = new SOME_AR_ENGINE();
while(true) {
  await nextVideoFrameReady();
  const {position, rotation} = ar_engine.computeObjectPose(video);
  some_object.position = position;
  some_object.rotation = rotation;
}
```

The tracked image can be of any origin, but it must be prepared: if it contains unnecessary elements, they should be removed.

To recognise an image using the MindAR library, you need to select landmarks on the image – the elements that will be used for recognition. This can be done using the image compiler available at https://hiukim.github.io/mind-ar-js-doc/tools/compile. Compiling results in the binary file `targets.mind`, which describes the reference points to be tracked.

Other libraries have similar means of obtaining image descriptions, often called NFT (natural feature tracking) marker compilers. Such an image should be visually complex and have a high resolution (details matter here). A visually complex image provides the software many opportunities to track unique and easily recognisable parts of the image.

The physical size of the NFT marker also affects the quality of its recognition: small images should be approached by the mobile device, while large ones should be kept away from it.

The recognition quality also depends on the brightness of the mobile device's screen; low-resolution cameras usually work better when they are close to the markers.

The Three.js library is a part of MindAR, which significantly simplifies their interaction: the `MindARThree` class constructor creates the objects necessary for working with Three.js – renderer, scene, and camera, which are available as `renderer`, `scene` and `camera` fields, respectively.

The anchor objects returned by the call to the `addAnchor` method, whose parameter corresponds to the number of the image to be recognised, are used to track target images and provide the position where the object should be placed.

Instead of adding Three.js objects directly to the scene, they are added to an anchor component – a `group` object of the `THREE.Group`class that defines a set of related objects whose position, orientation, and visibility can be controlled together. This anchor group is managed by the MindAR library, which will continuously update the group's position and orientation in accordance with our tracking set.

The `start` method of the `MindARThree` class sets up the parameters, turns on the camera, and loads all the necessary data into the web browser's memory.

For the renderer, camera, and scene to work, you must create a function to render them. In the unnamed callback function created by the `setAnimationLoop` function, for each frame, the `render` method is called from the `renderer` object, whose parameters are the `scene` and `camera` objects – this is the animation on the canvas.

The result is a fully functional WebAR application that tracks a single image (figure 8).

4. Methodology for developing WebAR applications for face tracking

4.1. Model of facial anchor points

The MindAR library has two main sets of modules – for working with images (`image`) and for working with faces (`face`).

The similarities between the image-tracking and face-tracking APIs are visible in the MindAR code. Despite the similarities, the `addAnchor` method interprets the parameter differently. For image tracking, it was the number of the target image; for face recognition, it is the number of the face reference point.

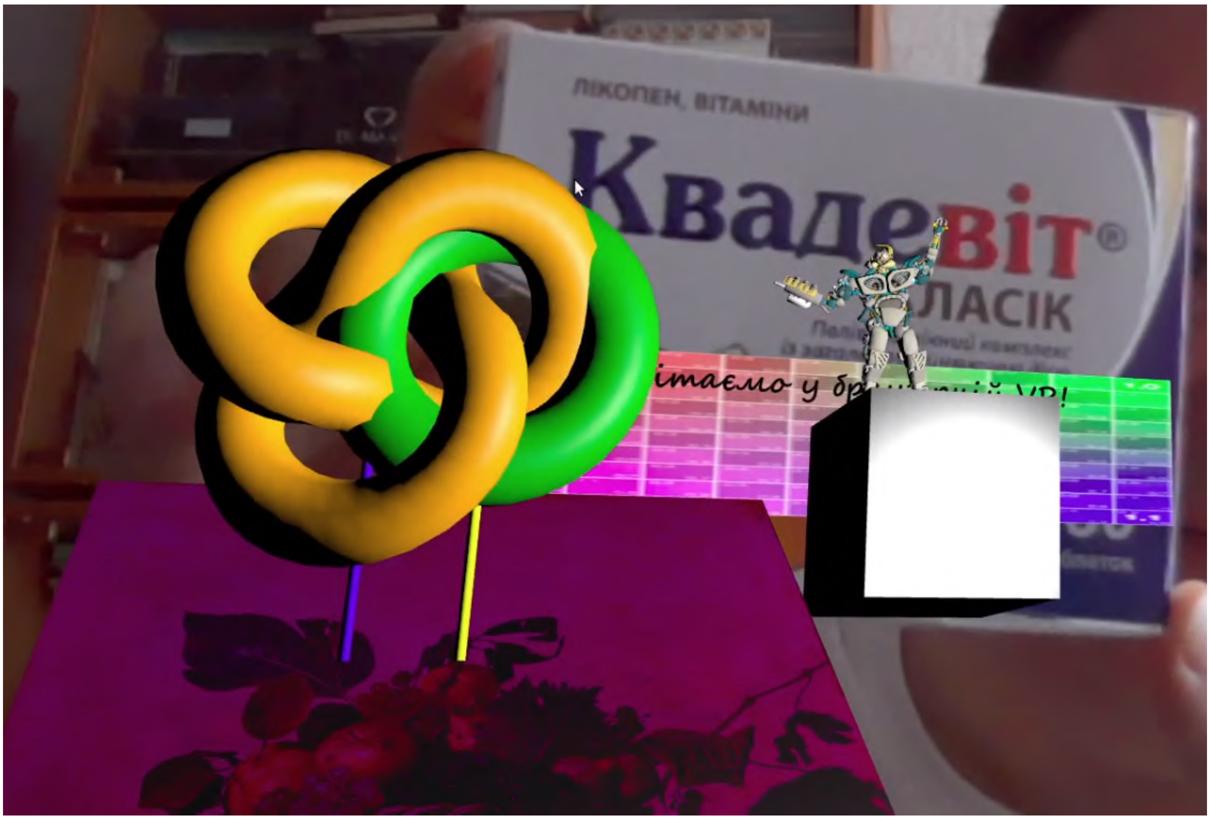


Figure 8: The result of image recognition.

Face landmark detection is based on the well-known TensorFlow library model [111]. MediaPipe Face Mesh model [112] is a convolutional neural network that detects 468 three-dimensional landmarks on the face (https://github.com/tensorflow/tfjs-models/raw/master/face-landmarks-detection/mesh_map.jpg), and we can bind objects to any of them (figure 9).

4.2. Putting a mesh on your face

A face mesh is another type of augmented reality that overlaps images (textures) on all the reference points of a person's face rather than being linked to individual points. Face meshes are used to create various makeup effects, tattoos, etc. – up to full face virtualisation.

The face mesh is not a predefined 3D model – it is dynamically generated with constant geometry updates.

To apply the mesh to the face, we need a suitable texture.

The mesh is created by calling `addFaceMesh`. The `addFaceMesh` method is similar in form to `addAnchor`, but they are different: `addAnchor` creates an empty group to which objects whose position is controlled by MindAR are added, while the `faceMesh` returned by `addFaceMesh` is a single displayed object whose geometry changes in each frame.

The material of the face mesh can be any texture – if you do not set it, the face mesh will look like the one shown in the first image (figure 10).

You can see the structure of this mesh in the second image (figure 10) – to do this, set the `wireframe` attribute of the image material.

The third and fourth images (figure 10) are examples of the modified texture of facial landmarks. In the documentation for Meta Spark Studio [113] you can find a set of textures for face meshes that can be used to create your mesh, as described in [114].

Creating a beautiful mesh requires specific artistic skills, but using the canonical texture (figure 9) is quite simple – apply the desired image over it and remove unnecessary lines.

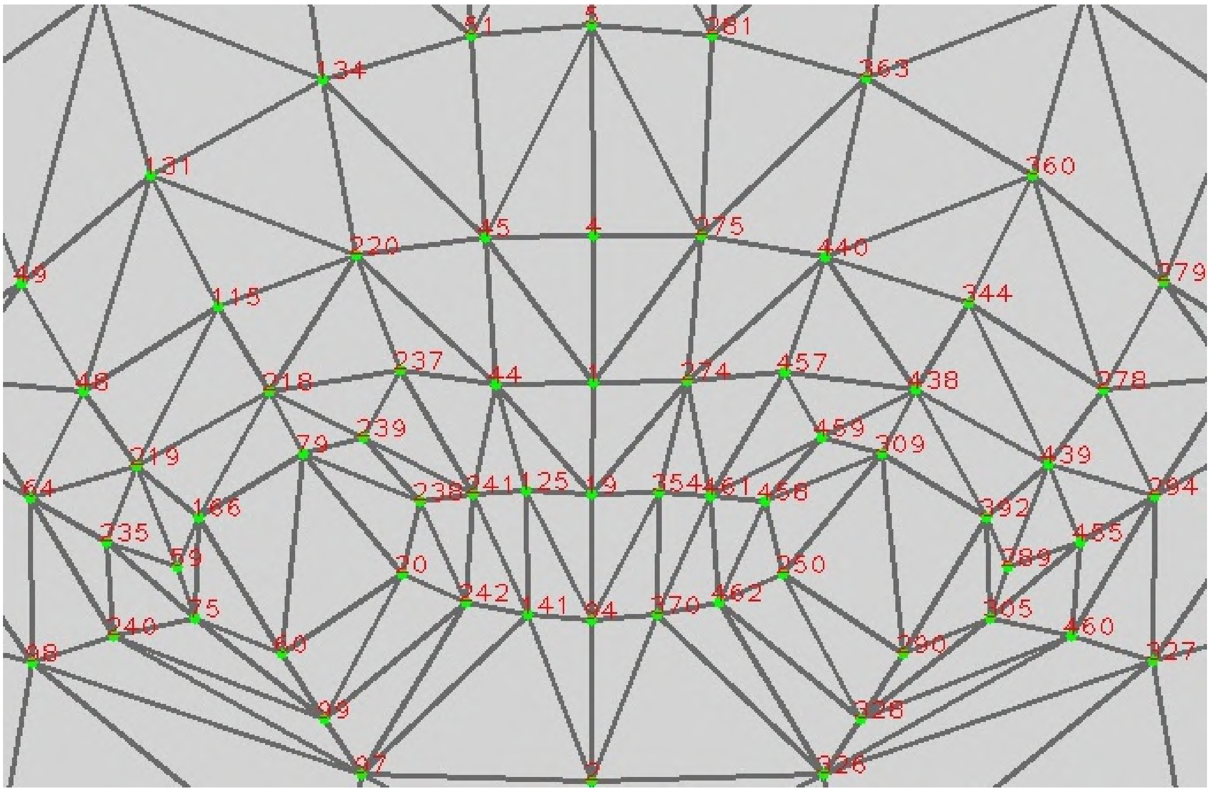


Figure 9: The reference points of the face (fragment).

5. Methodology for integrating machine learning models into WebAR applications

5.1. Integration of standard models

For machine learning on the Internet, TensorFlow [115] is the most commonly used free and open-source machine learning library developed by Google. Currently, it supports many languages, including major ones – Python, Java, and C++ – and community-supported ones: Haskell, C#, Julia, R, Ruby, Rust, and Scala. It is available on many platforms, including Linux, Windows, Android, and embedded platforms – the TensorFlow Lite library version is designed to run machine learning models on mobile devices, microcontrollers, IoT devices, etc.

TensorFlow.js [9] is a JavaScript version of TensorFlow that allows you to develop and use models using this language directly in the browser.

TensorFlow.js has many pre-trained models that can be used immediately [116]. A complete list of currently available models can be found at https://github.com/tensorflow/tfjs-models – many of them are extremely useful and can be an excellent addition to AR applications. If the required functionality is unavailable, you can create and train your models or modify existing ones.

TensorFlow.js is part of the MindAR library. However, models are not part of Tensorflow.js, so they need to be connected separately – as shown in the example of the `handpose.js` model described in [117]. This model is used to define the hand and its components.

The `handpose` model is loaded from the TensorFlow Hub (since 2023, a part of Kaggle) [118]: looking at this model repository, you can see that they take up a considerable amount of space, so the `load` method that loads them is called as an asynchronous function.

The `handpose` model processes individual frames taken from the video stream. This is a rather computationally intensive procedure, so, given that, as long as high accuracy of hand identification is not required, you can try to detect them not in every frame. The `detect` function creates a separate animation loop, in which for every tenth frame, the `estimateHands` method of the loaded model is

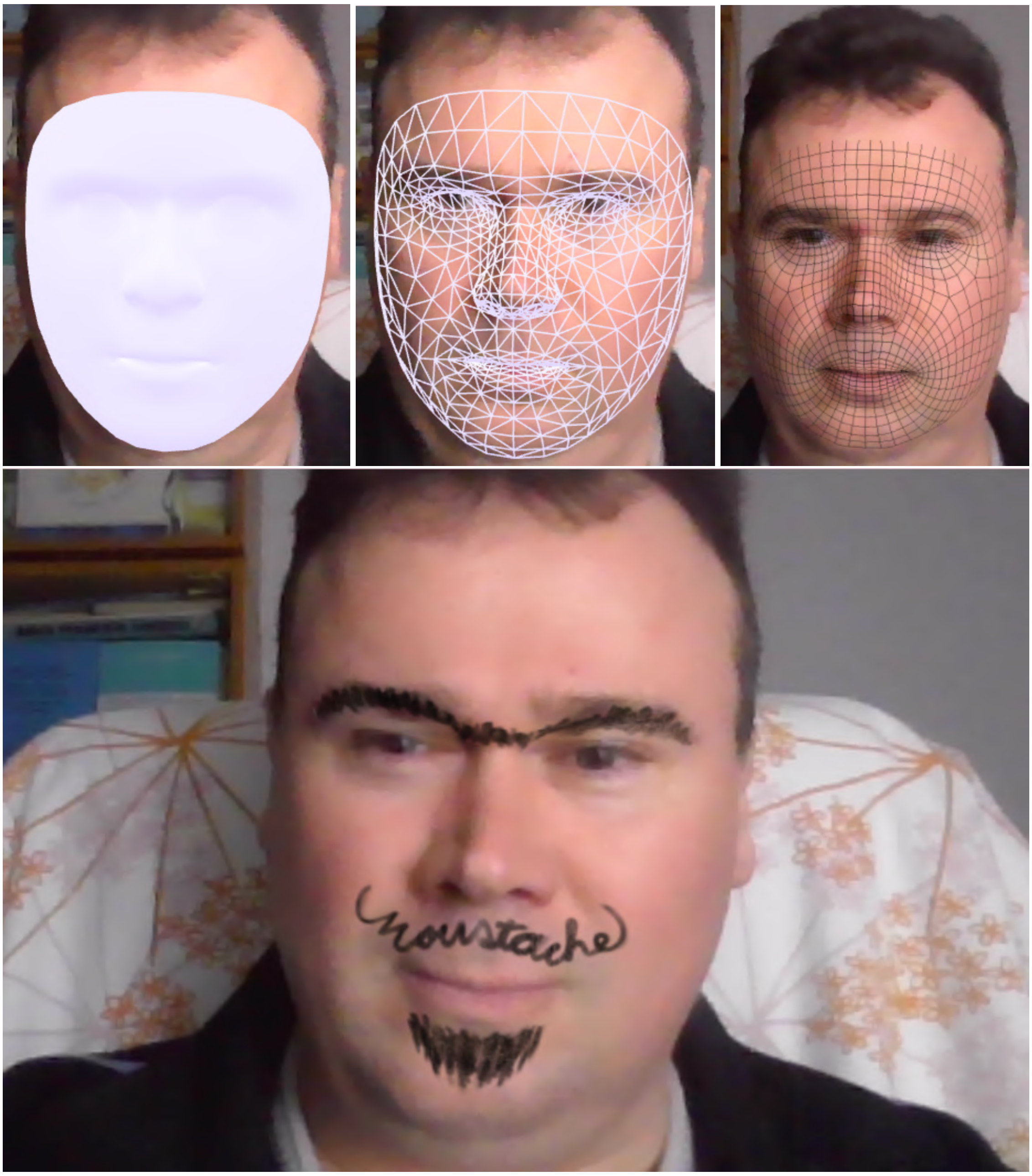


Figure 10: Face meshes.

called, and a `video` frame is passed to it. The method returns a `predictions` containing information about the hand images detected in the frame, so a non-zero array size is a sign that there was a hand in the frame:

```
const video = mindarThree.video;

let frameCount = 1;

const detect = async () => {
  if (frameCount % 10 == 0) {
    const predictions = await model.estimateHands(video);
  if (predictions.length > 0) {
    //...
```

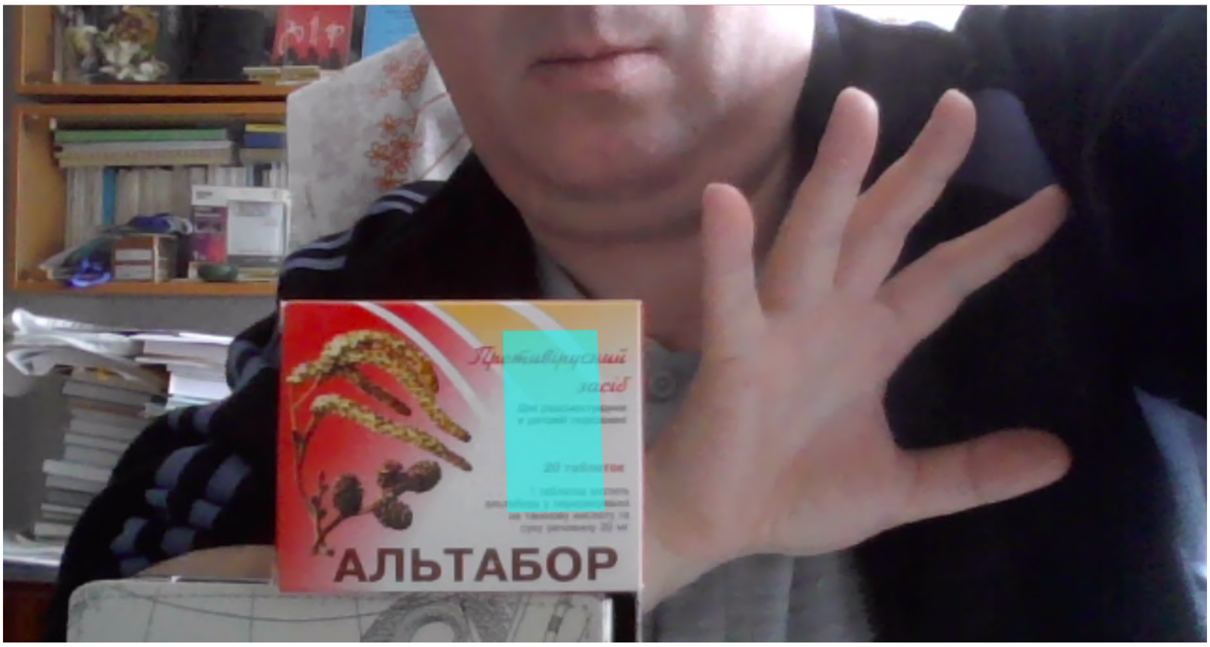


Figure 11: Gesture control of the size and position of a virtual object.

```
  }
  frameCount++;
  window.requestAnimationFrame(detect);
}

window.requestAnimationFrame(detect);
```

figure 11 shows an example of setting the position of the plane in the detected image so that it reflects the position of the bounding box of the hand in the frames – the effect is quite simple, but it provides an idea of how to use machine learning models in AR applications.

5.2. Developing custom models

To quickly create and train your model, you can use the Teachable Machine [119] is a part of the Google AI Experiment project (https://labs.google/ and https://experiments.withgoogle.com/), which allows building models to solve problems of image, sound, and pose classification.

To use the Teachable Machine, students are asked to create a new Google account or use an existing one, and then they can choose the type of model they want to create. There are three types of models available:

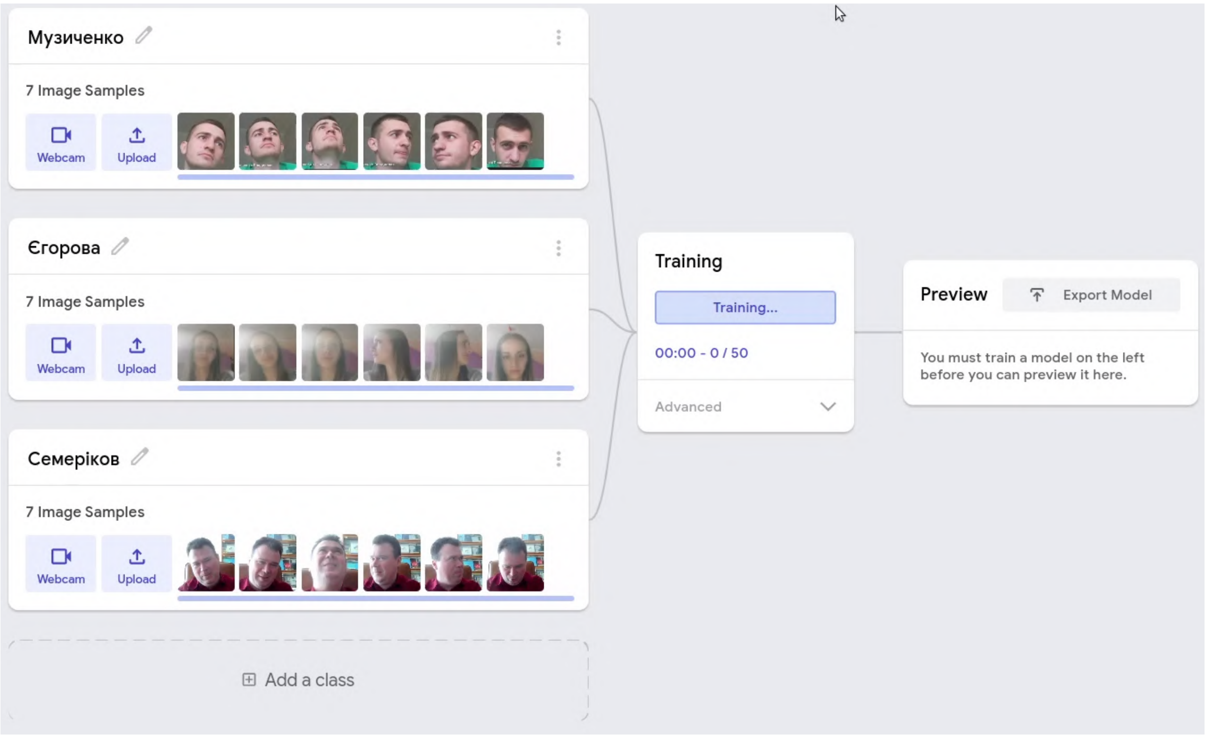
- Image recognition model allows you to identify objects in photos;
- Sound recognition model allows you to recognise audio recordings;
- Pose recognition model allows you to recognise body movements.

After selecting the model type, you need to provide data for training it through photos, audio recordings, or videos. Once the data is provided, the Teachable Machine will start training the model, which may take some time, depending on the size and complexity of the training. Once the model is trained, it is advisable to test it to ensure it correctly recognises the data. If the model is inaccurate enough, you can provide additional data to improve it. Once the model has been successfully trained and validated, it can be exported to other projects.

With Teachable Machine's wide range of features, we can recognise sounds, poses, faces, or any image. Nevertheless, to start using it, you must prepare photos and audio recordings for further experiments, train the selected model, and apply it directly to the web environment.

Clicking on the `Get Started` button on the home page will take you to a new window where you can use a project template or create your own.

When creating your project, choose which model you want to use. We choose `Image Project` and click on `Standard image model`. As a source of images, we suggest that students use their webcams and take a series of headshots from different angles (tilt and rotation angles), which we save in a pre-prepared catalogue. We will take several different images from each participant in the experiment and divide them into classes, noting the corresponding names (figure 12).



You must train a model on the left before you can preview it here.

Figure 12: Distribution of images by class.

For each image class, there is a probability that a particular image belongs to that class. Students can configure additional training parameters, such as the number of iterations and the model's learning speed.

Next, we move on to training the model – at this stage, all images are converted to the corresponding numerical tensors. The last step is to experiment by choosing images of different people (not just the participants in the experiment) and discussing the recognition results (figure 13).

5.3. Integration of custom models

The libraries included in Teachable Machine are based on TensorFlow models: MobileNet for image classification [120], Speech Commands for sound classification[121], and PoseNet for body pose classification [122].

Accordingly, the built face classification model can be exported and used the same way as the previously used models of facial landmarks and hand pose.

Clicking the `Export Model` button allows you to export in various formats:

- TensorFlow.js – placement of the model at https://teachablemachine.withgoogle.com/models/[...] or downloading the model and the JavaScript and p5.js code (figure 14);

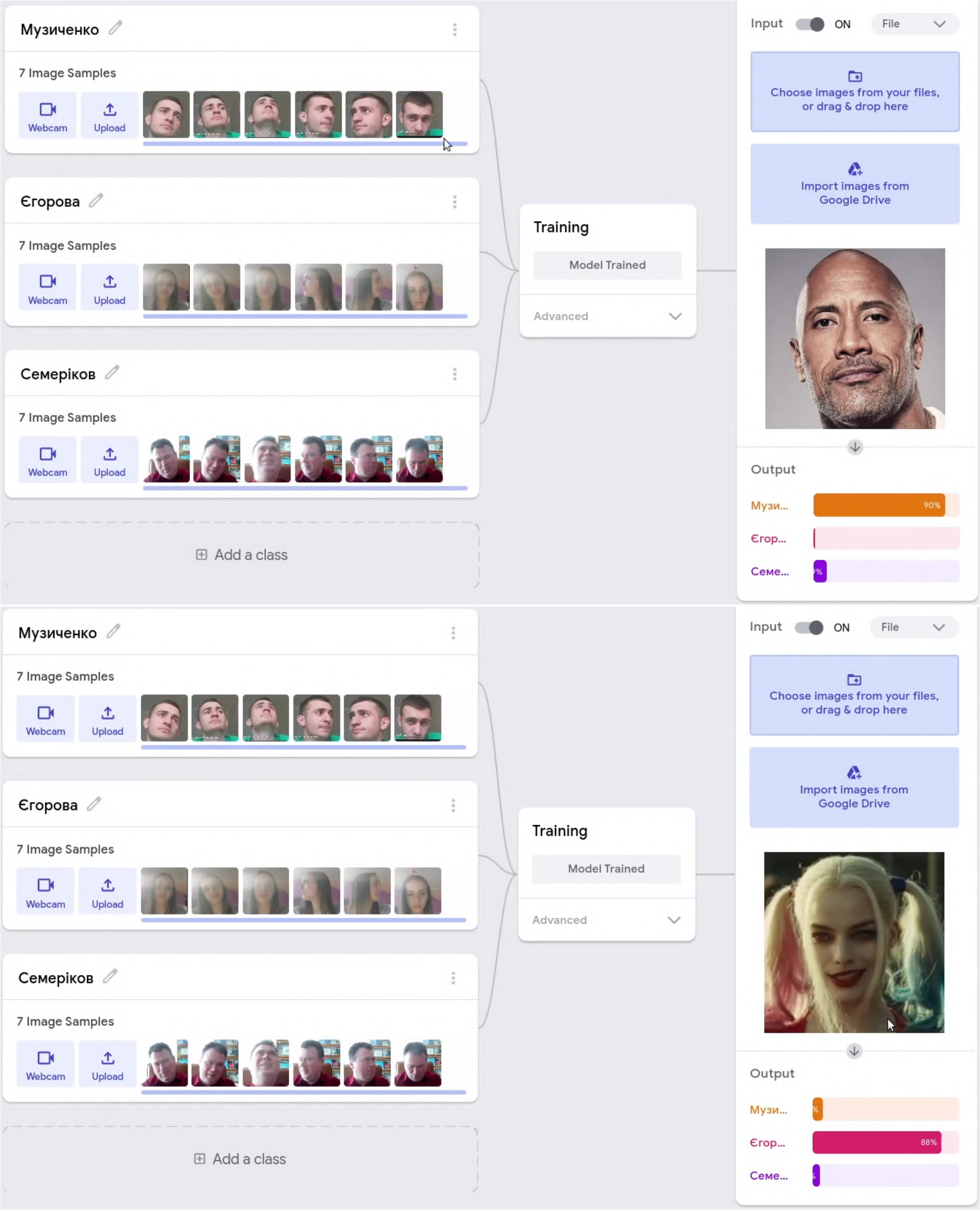
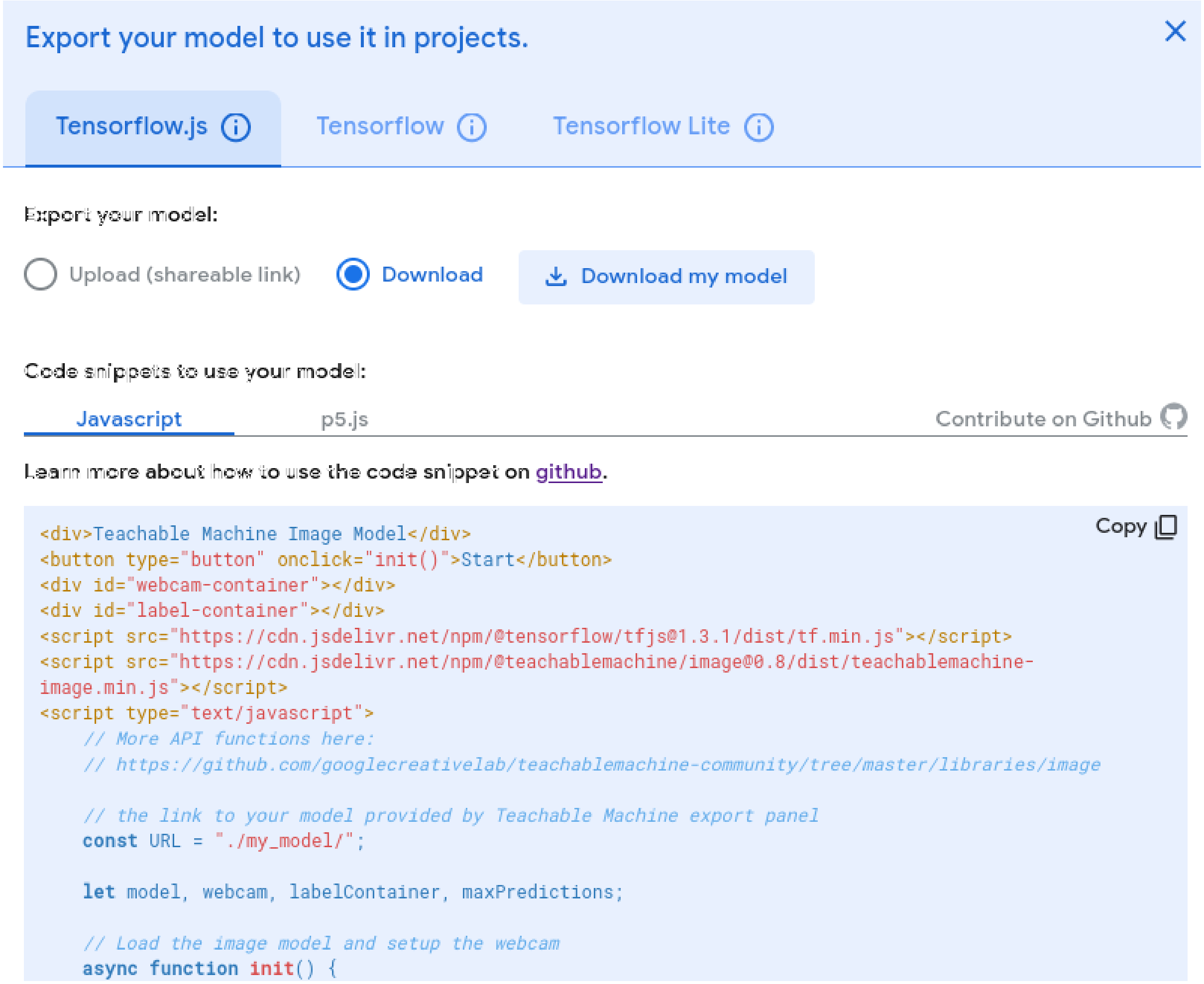


Figure 13: Results of the image recognition model.

- TensorFlow – download Python code and model in h5 (Keras) and Savedmodel (TensorFlow) formats;
- TensorFlow Lite – downloading a model in tflite format for IoT devices based on Android and Coral.

The archive with the model for TensorFlow.js contains three files:

- `metadata.json` – a text file in JSON format containing information about the version numbers of



Export your model to use it in projects.

Export your model:

○ Upload (shareable link) ● Download

Code snippets to use your model:

Learn more about how to use the code snippet on github.

```
<div>Teachable Machine Image Model</div>
<button type="button" onclick="init()">Start</button>
<div id="webcam-container"></div>
<div id="label-container"></div>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.3.1/dist/tf.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@teachablemachine/image@0.8/dist/teachablemachine-
image.min.js"></script>
<script type="text/javascript">
    // More API functions here:
    // https://github.com/googlecreativelab/teachablemachine-community/tree/master/libraries/image

    // the link to your model provided by Teachable Machine export panel
    const URL = "./my_model/";

    let model, webcam, labelContainer, maxPredictions;

    // Load the image model and setup the webcam
    async function init() {
```

Figure 14: Exporting the model for TensorFlow.js.

TensorFlow.js (`tfjsVersion`), Teachable Machine (`tmVersion`), libraries from the Teachable Machine (`packageVersion`) and its name (`packageName` – in our case, it is `@teachablemachine/image`), date of creation (`timeStamp`) and model name (`modelName` – by default `tm-my-image-model`), image size (`imageSize` – all images are resized to the same size) and categories (`labels`) used for data labelling;

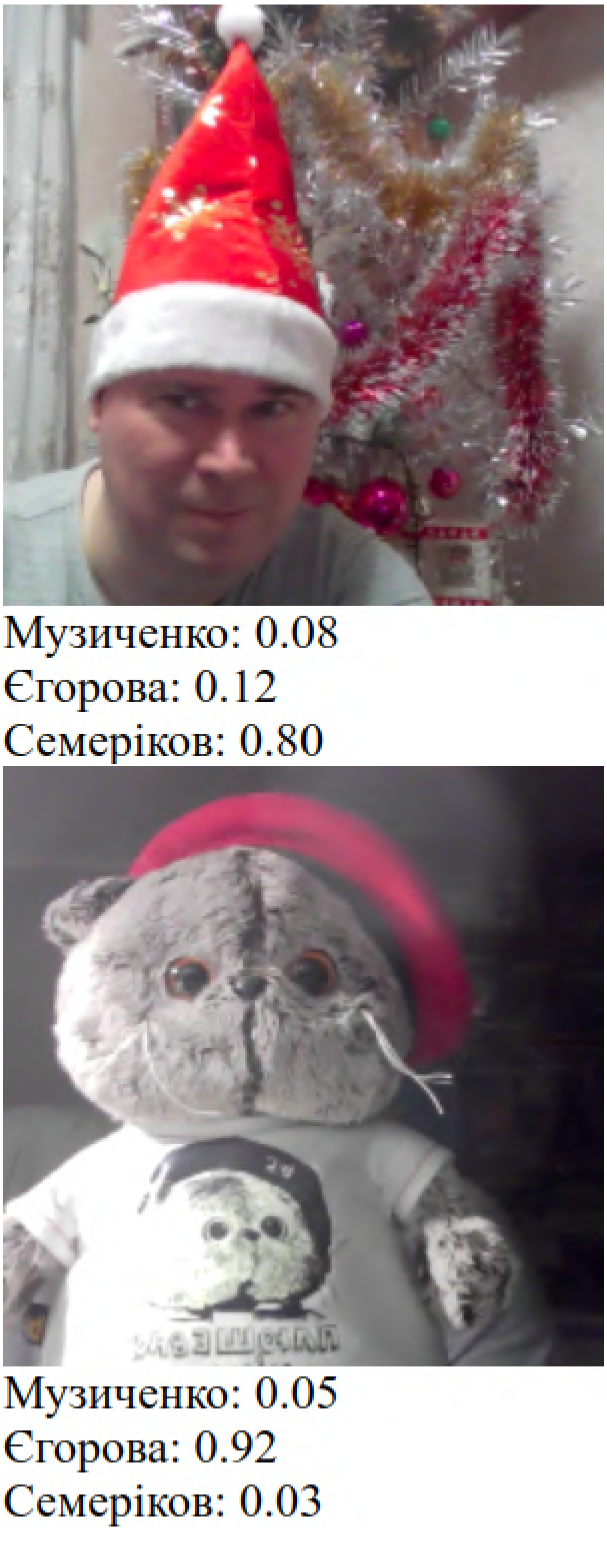
- `model.json` – a text file in JSON format containing information about the neural network architecture (`modelTopology`);
- `weights.bin` – a binary file containing the weighting coefficients of the neural network.

When exporting models, a test code is offered to verify them, from which you can learn how to connect the `tmImage` library and load the model by calling `load`, the parameters of which are the paths to the model architecture and metadata files – `model.json` and `metadata.json`.

After loading the model by calling the `getTotalClasses` method, you can determine the number of categories that the model will distinguish – in our case, this value, stored in `maxPredictions`, is three.

Just as before, every tenth frame is passed to the model for analysis by calling `predict`, which returns an array of two objects containing information about the category (`className`) and the probability that the image belongs to it (`probability`) – a string with information about them and is visualised.

From figure 15, we can see that the image on the left is identified correctly despite the change in the background compared to the training set (figure 12), while the image on the right is identified incorrectly.



Музиченко: 0.08
Єгорова: 0.12
Семеріков: 0.80

Музиченко: 0.05
Єгорова: 0.92
Семеріков: 0.03

Figure 15: Implementation of face recognition.

6. Conclusions

The completed solution to the problem of developing a methodology for teaching the development of augmented reality for the Web with integrated machine learning models made it possible to draw the following conclusions:

1. The bibliometric analysis of sources from the Crossref (19 documents in 2017-2022) and Scopus

(93 documents in 2001-2023) databases made it possible to identify the main concepts of the study, grouped into 4 clusters:

   a) The first cluster connects the *basic concepts of augmented reality in education*: augmented and virtual reality with education (including medical education) and human learning, including the use of smartphones;
   b) The second cluster links the *concepts of learning environment design*: teaching, engineering education, computer-aided instruction, e-learning, students, mobile learning, learning environments, user interfaces, education computing and curricula;
   c) The third cluster connects the *concepts of immersive learning environment implementation*: websites, motivation, learning systems, interactive learning environments, augmented reality applications and augmented reality technology;
   d) The fourth cluster contains the concept of *distance education*, linked in the first cluster to the concepts of augmented and virtual reality and the concept of education (including medical education), in the second to the concepts of students, engineering education, education computing, e-learning and curricula, and in the third to the concept of websites.

   The analysis of the distribution of concepts by the density of links and time made it possible to date the emergence of different concepts and track their development from educational applications of user interfaces to their augmentation.

2. The selected tools for developing augmented reality for the Web form three groups:
   a) fixed assets:
      - Simple Web Server provides the full functionality you need without installation needs that meet the requirements of simplicity and mobility;
      - ngrok traffic redirection allows access to a web server located behind a firewall (on a student's or teacher's computer), which creates conditions for working together remotely;
      - RemoteJS remote debugger allows you to debug JavaScript applications on mobile devices using desktop browsers;
   b) Three.js graphics library is a high-level implementation of the cross-platform WebGL display standard in JavaScript, which allows working with high-level graphical abstractions;
   c) MindAR augmented reality library allows working with natural images as augmented reality anchors and includes the Three.js and TensorFlow.js libraries – the latter is key for integrating machine learning models created with TensorFlow with WebAR applications built with MindAR.
3. In the process of developing and testing the methodology for developing WebAR applications for face tracking, the expediency of joint use of the MediaPipe Face Mesh model, a convolutional neural network that identifies 468 three-dimensional landmarks on the face, and the MindAR library, which allows to define any of them as an anchor, is substantiated. It is shown that the complete application of the MediaPipe Face Mesh model in the MindAR library is implemented in the form of a face mesh that is dynamically generated with constant geometry updates – a type of augmented reality associated with the overlay of images on all anchor points of the human face. Examples of using face meshes to create makeup effects, tattoos, etc., are presented.
4. The methodology of integrating machine learning models into WebAR applications involves mastering three main steps:
   a) The first step, *integration of standard models*, involves familiarisation with pre-trained TensorFlow.js models that can be used in WebAR applications. The article shows the feasibility of considering the handpose.js model used to determine the hand and its components, demonstrates the main problem of WebAR – a significant performance drop when applying the model to each frame, and suggests a way to solve it. As a result of the first step, a WebAR application for gestural size control is created and the position of the virtual object;
   b) The second step, *custom model development*, involves creating and training your TensorFlow models using the Teachable Machine, which allows you to build models to solve problems of image, sound, and pose classification;

c) The third step, *integration of user models*, is performed by exporting the face classification model built with the Teachable Machine and modifying the WebAR application developed in the first step: we load our model, determine the number of categories it will classify, and the object of augmented reality is information about each category and the probability that the webcam image belongs to it. The latter provides an opportunity to discuss the issues of classification errors and their dependence on both the settings of the model training parameters and the way the test images are presented to the WebAR application.

This study does not exhaust all the components of the problem, and further research is needed:

- history and prospects of WebAR development in education;
- a methodology for the joint use of different neural network modelling environments;
- development of WebAR libraries, in particular, in the direction of implementing ubiquitous augmented reality;
- the relationship between real and virtual in training in the context of a pandemic, natural disaster and military conflict.

References

[1] D. S. Shepiliev, S. O. Semerikov, Y. V. Yechkalo, V. V. Tkachuk, O. M. Markova, Y. O. Modlo, I. S. Mintii, M. M. Mintii, T. V. Selivanova, N. K. Maksyshko, T. A. Vakaliuk, V. V. Osadchyi, R. O. Tarasenko, S. M. Amelina, A. E. Kiv, Development of career guidance quests using WebAR, Journal of Physics: Conference Series 1840 (2021) 012028. URL: https://doi.org/10.1088/1742-6596/1840/1/012028. doi:10.1088/1742-6596/1840/1/012028.

[2] D. S. Shepiliev, Y. O. Modlo, Y. V. Yechkalo, V. V. Tkachuk, M. M. Mintii, I. S. Mintii, O. M. Markova, T. V. Selivanova, O. M. Drashko, O. O. Kalinichenko, T. A. Vakaliuk, V. V. Osadchyi, S. O. Semerikov, WebAR development tools: An overview, CEUR Workshop Proceedings 2832 (2020) 84–93. URL: http://ceur-ws.org/Vol-2832/paper12.pdf.

[3] O. V. Syrovatskyi, S. O. Semerikov, Y. O. Modlo, Y. V. Yechkalo, S. O. Zelinska, Augmented reality software design for educational purposes, CEUR Workshop Proceedings 2292 (2018) 193–225. URL: http://ceur-ws.org/Vol-2292/paper20.pdf.

[4] M. I. Striuk, A. M. Striuk, S. O. Semerikov, Mobility in the information society: a holistic model, Educational Technology Quarterly 2023 (2023) 277–301. URL: https://doi.org/10.55056/etq.619.

[5] AR.js Documentation, 2024. URL: https://ar-js-org.github.io/AR.js-Docs/.

[6] S. O. Semerikov, M. M. Mintii, I. S. Mintii, Review of the course "Development of Virtual and Augmented Reality Software" for STEM teachers: implementation results and improvement potentials, in: S. H. Lytvynova, S. O. Semerikov (Eds.), Proceedings of the 4th International Workshop on Augmented Reality in Education (AREdu 2021), Kryvyi Rih, Ukraine, May 11, 2021, volume 2898 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 159–177. URL: http://ceur-ws.org/Vol-2898/paper09.pdf.

[7] H. Yuen, HiuKim Yuen, 2023. URL: https://www.youtube.com/channel/UC-JyA1Z1-p0wgxj5WEX56wg/featured.

[8] H. Yuen, MindAR, 2023. URL: https://hiukim.github.io/mind-ar-js-doc/.

[9] TensorFlow.js | Machine Learning for JavaScript Developers, 2024. URL: https://www.tensorflow.org/js.

[10] Centre for Science and Technology Studies, Leiden University, The Netherlands, VOSviewer - Visualizing scientific landscapes, 2024. URL: https://www.vosviewer.com/.

[11] V. V. Tkachuk, S. O. Semerikov, Y. V. Yechkalo, O. M. Markova, M. M. Mintii, WebAR development tools: comparative analysis, Physical and Mathematical Education (2020). URL: https://doi.org/10.31110%2F2413-1571-2020-024-2-021. doi:10.31110/2413-1571-2020-024-2-021.

[12] J. An, L.-P. Poly, T. A. Holme, Usability testing and the development of an augmented reality application for laboratory learning, Journal of Chemical Education 97 (2020) 97–105. URL: https://doi.org/10.1021/acs.jchemed.9b00453.

[13] P. E. Antoniou, E. Dafli, G. Arfaras, P. D. Bamidis, Versatile mixed reality medical educational spaces; requirement analysis from expert users, Personal and Ubiquitous Computing 21 (2017) 1015–1024. URL: https://doi.org/10.1007/s00779-017-1074-5.

[14] J. V. Arteaga, M. L. Gravini-Donado, L. D. Z. Riva, Digital technologies for heritage teaching: Trend analysis in new realities, International Journal of Emerging Technologies in Learning 16 (2021) 132–148. URL: https://doi.org/10.3991/ijet.v16i21.25149.

[15] T. N. Arvanitis, A. Petrou, J. F. Knight, S. Savas, S. Sotiriou, M. Gargalakos, E. Gialouri, Human factors and qualitative pedagogical evaluation of a mobile augmented reality system for science education used by learners with physical disabilities, Personal and Ubiquitous Computing 13 (2009) 243–250. URL: https://doi.org/10.1007/s00779-007-0187-7.

[16] H. T. Atmaca, O. S. Terzi, Building a web-augmented reality application for demonstration of kidney pathology for veterinary education, Polish Journal of Veterinary Sciences 24 (2021) 345–350. URL: https://doi.org/10.24425/pjvs.2021.137671.

[17] Y. Baashar, G. Alkawsi, W. N. W. Ahmad, H. Alhussian, A. Alwadain, L. F. Capretz, A. Babiker, A. Alghail, Effectiveness of using augmented reality for training in the medical professions: Meta-analysis, JMIR Serious Games 10 (2022) e32715. URL: https://doi.org/10.2196/32715.

[18] K. Bhavika, J. Martin, B. Ardit, Technology will never replace hands on surgical training in plastic surgery, Journal of Plastic, Reconstructive and Aesthetic Surgery 75 (2022) 439–488. URL: https://doi.org/10.1016/j.bjps.2021.11.034.

[19] H. M. Bradford, C. L. Farley, M. Escobar, E. T. Heitzler, T. Tringali, K. C. Walker, Rapid curricular innovations during covid-19 clinical suspension: Maintaining student engagement with simulation experiences, Journal of Midwifery and Women's Health 66 (2021) 366–371. URL: https://doi.org/10.1111/jmwh.13246.

[20] A. Brunzini, A. Papetti, E. B. Serrani, M. Scafà, M. Germani, How to Improve Medical Simulation Training: A New Methodology Based on Ergonomic Evaluation, in: W. Karwowski, T. Ahram, S. Nazir (Eds.), Advances in Human Factors in Training, Education, and Learning Sciences, volume 963 of *Advances in Intelligent Systems and Computing*, Springer International Publishing, Cham, 2020, pp. 145–155. URL: https://doi.org/10.1007/978-3-030-20135-7_14.

[21] B. K. Burian, M. Ebnali, J. M. Robertson, D. Musson, C. N. Pozner, T. Doyle, D. S. Smink, C. Miccile, P. Paladugu, B. Atamna, S. Lipsitz, S. Yule, R. . Dias, Using extended reality (xr) for medical training and real-time clinical support during deep space missions, Applied Ergonomics 106 (2023) 103902. URL: https://doi.org/10.1016/j.apergo.2022.103902.

[22] I. Coma-Tatay, S. Casas-Yrurzum, P. Casanova-Salas, M. Fernández-Marín, Fi-ar learning: a web-based platform for augmented reality educational content, Multimedia Tools and Applications 78 (2019) 6093–6118. URL: https://doi.org/10.1007/s11042-018-6395-5.

[23] F. Cortés Rodríguez, M. Dal Peraro, L. Abriata, Online tools to easily build virtual molecular models for display in augmented and virtual reality on the web, Journal of Molecular Graphics and Modelling 114 (2022) 108164. URL: https://doi.org/10.1016/j.jmgm.2022.108164.

[24] T. Coughlin, Impact of covid-19 on the consumer electronics market, IEEE Consumer Electronics Magazine 10 (2021) 58–59. URL: https://doi.org/10.1109/MCE.2020.3016753.

[25] P. G. Crandall, R. K. Engler III, D. E. Beck, S. A. Killian, C. A. O'Bryan, N. Jarvis, E. Clausen, Development of an augmented reality game to teach abstract concepts in food chemistry, Journal of Food Science Education 14 (2015) 18–23. URL: https://ift.onlinelibrary.wiley.com/doi/abs/10.1111/1541-4329.12048. doi:`https://doi.org/10.1111/1541-4329.12048`. `arXiv:https://ift.onlinelibrary.wiley.com/doi/pdf/10.1111/1541-4329.12048`.

[26] S. A. Dar, Mobile library initiatives: a new way to revitalize the academic library settings, Library Hi Tech News 36 (2019) 15–21. URL: https://doi.org/10.1108/LHTN-05-2019-0032.

[27] L. Dunkel, L. Fernandez-Luque, S. Loche, M. O. Savage, Digital technologies to improve the precision of paediatric growth disorder diagnosis and management, Growth Hormone and IGF

Research 59 (2021) 101408. URL: https://doi.org/10.1016/j.ghir.2021.101408.

[28] E. Erçağ, A. Yasakcı, The perception scale for the 7e model-based augmented reality enriched computer course (7emagbaÖ): Validity and reliability study, Sustainability 14 (2022) 12037. URL: https://doi.org/10.3390/su141912037.

[29] E. Faridi, A. Ghaderian, F. Honarasa, A. Shafie, Next generation of chemistry and biochemistry conference posters: Animation, augmented reality, visitor statistics, and visitors' attention, Biochemistry and Molecular Biology Education 49 (2021) 619–624. URL: https://doi.org/10.1002/bmb.21520.

[30] S. Farra, E. Hodgson, E. Miller, N. Timm, W. Brady, M. Gneuhs, J. Ying, J. Hausfeld, E. Cosgrove, A. Simon, M. Bottomley, Effects of virtual reality simulation on worker emergency evacuation of neonates, Disaster Medicine and Public Health Preparedness 13 (2019) 301–308. URL: https://doi.org/10.1017/dmp.2018.58.

[31] N. Gordon, M. Brayshaw, T. Aljaber, Heuristic Evaluation for Serious Immersive Games and M-instruction, in: P. Zaphiris, A. Ioannou (Eds.), Learning and Collaboration Technologies, volume 9753 of *Lecture Notes in Computer Science*, Springer International Publishing, Cham, 2016, pp. 310–319. URL: https://doi.org/10.1007/978-3-319-39483-1_29.

[32] B. Hensen, I. Koren, R. Klamma, A. Herrler, An augmented reality framework for gamified learning, in: G. Hancke, M. Spaniol, K. Osathanunkul, S. Unankard, R. Klamma (Eds.), Advances in Web-Based Learning – ICWL 2018, volume 11007 of *Lecture Notes in Computer Science*, Springer International Publishing, Cham, 2018, pp. 67–76. URL: https://doi.org/10.1007/978-3-319-96565-9_7.

[33] T. G. Hoog, L. M. Aufdembrink, N. J. Gaut, R.-J. Sung, K. P. Adamala, A. E. Engelhart, Rapid deployment of smartphone-based augmented reality tools for field and online education in structural biology, Biochemistry and Molecular Biology Education 48 (2020) 448–451. URL: https://doi.org/10.1002/bmb.21396.

[34] T.-C. Huang, Seeing creativity in an augmented experiential learning environment, Universal Access in the Information Society 18 (2019) 301–313. URL: https://doi.org/10.1007/s10209-017-0592-2.

[35] M. B. Ibáñez, Á. Di Serio, D. Villarán, C. Delgado Kloos, Experimenting with electromagnetism using augmented reality: Impact on flow student experience and educational effectiveness, Computers and Education 71 (2014) 1–13. URL: https://doi.org/10.1016/j.compedu.2013.09.004.

[36] M.-B. Ibanez, A. Di-Serio, D. Villaran-Molina, C. Delgado-Kloos, Augmented reality-based simulators as discovery learning tools: An empirical study, IEEE Transactions on Education 58 (2015) 208–213. URL: https://doi.org/10.1109/TE.2014.2379712.

[37] M. B. Ibáñez, J. Peláez, C. D. Kloos, Using an Augmented Reality Geolocalized Quiz Game as an Incentive to Overcome Academic Procrastination, in: M. E. Auer, T. Tsiatsos (Eds.), Mobile Technologies and Applications for the Internet of Things, volume 909 of *Advances in Intelligent Systems and Computing*, Springer International Publishing, Cham, 2019, pp. 175–184. URL: https://doi.org/10.1007/978-3-030-11434-3_21.

[38] M. Ibáñez, A. Uriarte Portillo, R. Zatarain Cabada, M. Barrón, Impact of augmented reality technology on academic achievement and motivation of students from public and private mexican schools. a case study in a middle-school geometry course, Computers and Education 145 (2020) 103734. URL: https://doi.org/10.1016/j.compedu.2019.103734.

[39] K. Jung, V. Nguyen, S.-C. Yoo, S. Kim, S. Park, M. Currie, Palmitoar: The last battle of the u.s. civil war reenacted using augmented reality, ISPRS International Journal of Geo-Information 9 (2020) 75. URL: https://doi.org/10.3390/ijgi9020075.

[40] B. Kang, J. Heo, H. H. S. Choi, K. H. Lee, 2030 toy web of the future, in: S. Kim, J.-W. Jung, N. Kubota (Eds.), Soft Computing in Intelligent Control, volume 272 of *Advances in Intelligent Systems and Computing*, Springer International Publishing, Cham, 2014, pp. 69–75. URL: https://doi.org/10.1007/978-3-319-05570-1_8.

[41] S. I. Karas, E. V. Grakova, M. V. Balakhonova, M. B. Arzhanik, E. E. Kara-Sal, Distance learning in cardiology: The use of multimedia clinical diagnostic tasks, Russian Journal of Cardiology 25 (2020) 187–194. URL: https://doi.org/10.15829/1560-4071-2020-4116.

[42] M. Karayilan, S. M. McDonald, A. J. Bahnick, K. M. Godwin, Y. M. Chan, M. L. Becker, Reassessing undergraduate polymer chemistry laboratory experiments for virtual learning environments, Journal of Chemical Education 99 (2022) 1877–1889. URL: https://doi.org/10.1021/acs.jchemed.1c01259.

[43] T. Katika, S. N. Bolierakis, E. Vasilopoulos, M. Antonopoulos, G. Tsimiklis, I. Karaseitanidis, A. Amditis, Coupling AR with Object Detection Neural Networks for End-User Engagement, in: G. Zachmann, M. Alcañiz Raya, P. Bourdot, M. Marchal, J. Stefanucci, X. Yang (Eds.), Virtual Reality and Mixed Reality, volume 13484 of *Lecture Notes in Computer Science*, Springer International Publishing, Cham, 2022, pp. 135–145. URL: https://doi.org/10.1007/978-3-031-16234-3_8.

[44] I. Kazanidis, N. Pellas, A. Christopoulos, A learning analytics conceptual framework for augmented reality-supported educational case studies, Multimodal Technologies and Interaction 5 (2021) 9. URL: https://doi.org/10.3390/mti5030009.

[45] H. Le, M. Nguyen, An Online Platform for Enhancing Learning Experiences with Web-Based Augmented Reality and Pictorial Bar Code, in: V. Geroimenko (Ed.), Augmented Reality in Education: A New Technology for Teaching and Learning, Springer Series on Cultural Computing, Springer International Publishing, Cham, 2020, pp. 45–57. URL: https://doi.org/10.1007/978-3-030-42156-4_3. doi:10.1007/978-3-030-42156-4_3.

[46] E. Liu, S. Cai, Z. Liu, C. Liu, *WebART*: Web-based augmented reality learning resources authoring tool and its user experience study among teachers, IEEE Transactions on Learning Technologies 16 (2023) 53–65. URL: https://doi.org/10.1109/TLT.2022.3214854.

[47] D. Lou, Two fast prototypes of web-based augmented reality enhancement for books, Library Hi Tech News 36 (2019) 19–24. URL: https://doi.org/10.1108/LHTN-08-2019-0057.

[48] C. Lytridis, A. Tsinakos, I. Kazanidis, Artutor—an augmented reality platform for interactive distance learning, Education Sciences 8 (2018) 6. URL: https://doi.org/10.3390/educsci8010006.

[49] R. Marín, P. J. Sanz, A. P. Del Pobil, The uji online robot: An education and training experience, Autonomous Robots 15 (2003) 283–297. URL: https://doi.org/10.1023/A:1026220621431.

[50] D. R. Nemirovsky, A. J. Garcia, P. Gupta, E. Shoen, N. Walia, Evaluation of surgical improvement of clinical knowledge ops (sicko), an interactive training platform, Journal of Digital Imaging 34 (2021) 1067–1071. URL: https://doi.org/10.1007/s10278-021-00482-x.

[51] V. T. Nguyen, K. Jung, T. Dang, Blocklyar: A visual programming interface for creating augmented reality experiences, Electronics 9 (2020) 1–20. URL: https://doi.org/10.3390/electronics9081205.

[52] S. Brewster, R. Murray-Smith (Eds.), Haptic Human-Computer Interaction: First International Workshop, Glasgow, UK, August 31 - September 1, 2000, Proceedings, volume 2058 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin Heidelberg, 2001. URL: https://doi.org/10.1007/3-540-44589-7. doi:10.1007/3-540-44589-7.

[53] J. D. Westwood, S. W. Westwood, L. Felländer-Tsai, C. M. Fidopiastis, A. Liu, S. Senger, K. G. Vosburgh (Eds.), Medicine Meets Virtual Reality 22 - NextMed, MMVR 2016, Los Angeles, California, USA, April 7-9, 2016, volume 220 of *Studies in Health Technology and Informatics*, IOS Press, 2016. URL: http://ebooks.iospress.nl/volume/medicine-meets-virtual-reality-22-nextmed-mmvr22.

[54] W. Budiharto, A. A. S. Gunawan, L. A. Wulandhari, Williem, Faisal, R. Sutoyo, Meiliana, D. Suryani, Y. Arifin (Eds.), The 3rd International Conference on Computer Science and Computational Intelligence (ICCSCI 2018) : Empowering Smart Technology in Digital Era for a Better Life, volume 135 of *Procedia Computer Science*, Elsevier B.V., 2018. URL: https://www.sciencedirect.com/journal/procedia-computer-science/vol/135/suppl/C.

[55] L. Rønningsbakk, T.-T. Wu, F. E. Sandnes, Y.-M. Huang (Eds.), Innovative Technologies and Learning: Second International Conference, ICITL 2019, Tromsø, Norway, December 2–5, 2019, Proceedings, volume 11937 of *Lecture Notes in Computer Science*, Springer International Publishing, 2019. URL: https://doi.org/10.1007/978-3-030-35343-8. doi:10.1007/978-3-030-35343-8.

[56] Preface, Journal of Physics: Conference Series 1860 (2021) 011001. URL: https://doi.org/10.1088/1742-6596/1860/1/011001. doi:10.1088/1742-6596/1860/1/011001.

[57] N. Nordin, N. R. M. Nordin, W. Omar, Rev-opoly: A study on educational board game with webbased augmented reality, Asian Journal of University Education 18 (2022) 81–90. URL:

https://doi.org/10.24191/ajue.v18i1.17172.

[58] M. E. Rollo, E. J. Aguiar, R. L. Williams, K. Wynne, M. Kriss, R. Callister, C. E. Collins, Ehealth technologies to support nutrition and physical activity behaviors in diabetes self-management, Diabetes, Metabolic Syndrome and Obesity: Targets and Therapy 9 (2016) 381–390. URL: https://doi.org/10.2147/DMSO.S95247.

[59] C. Samat, S. Chaijaroen, Design and Development of Constructivist Augmented Reality (AR) Book Enhancing Analytical Thinking in Computer Classroom, in: L. Rønningsbakk, T.-T. Wu, F. E. Sandnes, Y.-M. Huang (Eds.), Innovative Technologies and Learning, volume 11937 of *Lecture Notes in Computer Science*, Springer International Publishing, Cham, 2019, pp. 175–183. URL: https://doi.org/10.1007/978-3-030-35343-8_19.

[60] S. M. E. Sepasgozar, Digital twin and web-based virtual gaming technologies for online education: A case of construction management and engineering, Applied Sciences 10 (2020) 4678. URL: https://doi.org/10.3390/app10134678.

[61] K. Sharp, M. McCorvie, M. Wagner, Sharing hidden histories: The xrchaeology at miller grove, a free african american community in southern illinois, Journal of African Diaspora Archaeology and Heritage 12 (2023) 5–31. URL: https://doi.org/10.1080/21619441.2021.1902706.

[62] E. Smith, K. McRae, G. Semple, H. Welsh, D. Evans, P. Blackwell, Enhancing vocational training in the post-covid era through mobile mixed reality, Sustainability 13 (2021) 6144. URL: https://doi.org/10.3390/su13116144.

[63] C. Thabvithorn, C. Samat, Development of Web-Based Learning with Augmented Reality (AR) to Promote Analytical Thinking on Computational Thinking for High School, in: Y.-M. Huang, S.-C. Cheng, J. Barroso, F. E. Sandnes (Eds.), Innovative Technologies and Learning, volume 13449 of *Lecture Notes in Computer Science*, Springer International Publishing, Cham, 2022, pp. 125–133. URL: https://doi.org/10.1007/978-3-031-15273-3_14.

[64] F. Turner, I. Welch, The mixed reality toolkit as the next step in the mass customization co-design experience, International Journal of Industrial Engineering and Management 10 (2019) 191–199. URL: https://doi.org/10.24867/IJIEM-2019-2-239.

[65] A. Vahabzadeh, N. Keshav, J. P. Salisbury, N. Sahin, Improvement of attention-deficit/hyperactivity disorder symptoms in school-aged children, adolescents, and young adults with autism via a digital smartglasses-based socioemotional coaching aid: Short-term, uncontrolled pilot study, JMIR Mental Health 5 (2018) e25. URL: https://doi.org/10.2196/mental.9631.

[66] D. Villarán, M. B. Ibáñez, C. D. Kloos, Augmented reality-based simulations embedded in problem based learning courses, in: G. Conole, T. Klobučar, C. Rensing, J. Konert, E. Lavoué (Eds.), Design for Teaching and Learning in a Networked World, volume 9307 of *Lecture Notes in Computer Science*, Springer International Publishing, Cham, 2015, pp. 540–543. URL: https://doi.org/10.1007/978-3-319-24258-3_55.

[67] S. Yang, B. Mei, X. Yue, Mobile augmented reality assisted chemical education: Insights from elements 4d, Journal of Chemical Education 95 (2018) 1060–1062. URL: https://doi.org/10.1021/acs.jchemed.8b00017.

[68] R. Zatarain-Cabada, M. Barrón-Estrada, B. A. Cárdenas-Sainz, M. E. Chavez-Echeagaray, Experiences of web-based extended reality technologies for physics education, Computer Applications in Engineering Education 31 (2023) 63–82. URL: https://doi.org/10.1002/cae.22571.

[69] N. U. Zitzmann, L. Matthisson, H. Ohla, T. Joda, Digital undergraduate education in dentistry: A systematic review, International Journal of Environmental Research and Public Health 17 (2020) 3269. URL: https://doi.org/10.3390/ijerph17093269.

[70] S. Hai-Jew, Adult Coloring Books as Emotional Salve/Stress Relief, Tactual-Visual Learning: An Analysis from Mass-Scale Social Imagery, in: Common Visual Art in a Social Digital Age, Nova Science Publishers, Inc., 2022, pp. 171–186.

[71] L. Huang, Chemistry Apps on Smartphones and Tablets, in: J. García-Martínez, E. Serrano-Torregrosa (Eds.), Chemistry Education, John Wiley & Sons, Ltd, 2015, pp. 621–650. URL: https://doi.org/10.1002/9783527679300.ch25.

[72] C. A. Jara, F. A. Candelas, F. Torres, Internet virtual and remote control interface for robotics

education, in: Developments in Higher Education, Nova Science Publishers, Inc., 2009, pp. 136–154.

[73] E. Redondo, I. Navarro, A. Sánchez, D. Fonseca, Implementation of Augmented Reality in "3.0 Learning" Methodology: Case Studies with Students of Architecture Degree, in: B. Pătruţ, M. Pătruţ, C. Cmeciu (Eds.), Social Media and the New Academic Environment: Pedagogical Challenges, IGI Global, Hershey, PA, 2013, pp. 391–413. URL: https://doi.org/10.4018/978-1-4666-2851-9.ch019.

[74] J. Al-Gharaibeh, C. Jeffery, Portable non-player character tutors with quest activities, in: 2010 IEEE Virtual Reality Conference (VR), 2010, pp. 253–254. URL: https://doi.org/10.1109/VR.2010.5444779.

[75] P. E. Antoniou, E. Dafli, G. Arfaras, P. D. Bamidis, Versatile Mixed Reality Educational Spaces; A Medical Education Implementation Case, in: N. Georgalas, Q. Jin, J. Garcia-Blas, J. Carretero, I. Ray (Eds.), Proceedings - 2016 15th International Conference on Ubiquitous Computing and Communications and 2016 8th International Symposium on Cyberspace and Security, IUCC-CSS 2016, Institute of Electrical and Electronics Engineers Inc., 2017, pp. 132–137. URL: https://doi.org/10.1109/IUCC-CSS.2016.026.

[76] S. Anwar, J. LeClair, A. Peskin, Development Of Nanotechnology And Power Systems Options For An On Line Bseet Degree, in: 2010 Annual Conference & Exposition, ASEE Conferences, Louisville, Kentucky, 2010, pp. 15.420.1 – 15.420.10. URL: https://doi.org/10.18260/1-2--15776.

[77] B. Cardenas-Sainz, R. Zatarain-Cabada, M. Barron-Estrada, M. Chavez-Echeagaray, R. Cabada, FisicARtivo: Design of a learning tool for physics education using web-based XR technology, in: 2022 IEEE Mexican International Conference on Computer Science, ENC 2022 - Proceedings, Institute of Electrical and Electronics Engineers Inc., 2022. URL: https://doi.org/10.1109/ENC56672.2022.9882930.

[78] I. Demir, Interactive web-based hydrological simulation system as an education platform, in: A. E. Rizzoli, N. W. T. Quinn, D. P. Ames (Eds.), Proceedings - 7th International Congress on Environmental Modelling and Software: Bold Visions for Environmental Modeling, iEMSs 2014, volume 2, International Environmental Modelling and Software Society, 2014, pp. 910–912. URL: https://doi.org/10.17077/aseenmw2014.1008.

[79] M. Farella, D. Taibi, M. Arrigo, G. Todaro, G. Fulantelli, G. Chiazzese, An augmented reality mobile learning experience based on treasure hunt serious game, in: C. Busch, M. Steinicke, R. Friess, T. Wendler (Eds.), Proceedings of the European Conference on e-Learning, ECEL, Academic Conferences and Publishing International Limited, 2021, pp. 148–154. URL: https://doi.org/10.34190/EEL.21.109.

[80] J. Ferguson, M. Mentzelopoulos, A. Protopsaltis, D. Economou, Small and flexible web based framework for teaching QR and AR mobile learning application development, in: Proceedings of 2015 International Conference on Interactive Mobile Communication Technologies and Learning, IMCL 2015, Institute of Electrical and Electronics Engineers Inc., 2015, pp. 383–385. URL: https://doi.org/10.1109/IMCTL.2015.7359624.

[81] Harun, N. Tuli, A. Mantri, Experience fleming's rule in electromagnetism using augmented reality: Analyzing impact on students learning, Procedia Computer Science 172 (2020) 660–668. URL: https://doi.org/10.1016/j.procs.2020.05.086.

[82] T. Kobayashi, H. Sasaki, A. Toguchi, K. Mizuno, A discussion on web-based learning contents with the AR technology and its authoring tools to improve students' skills in exercise courses, in: A. F. Mohd Ayub, A. Kashihara, T. Matsui, C.-C. Liu, H. Ogata, S. C. Kong (Eds.), Work-In-Progress Poster - Proceedings of the 22nd International Conference on Computers in Education, ICCE 2014, Asia-Pacific Society for Computers in Education, 2014, pp. 34–36.

[83] L. O. Maggi, J. M. X. N. Teixeira, J. R. F. E. S. Junior, J. P. C. Cajueiro, P. V. S. G. De Lima, M. H. R. De Alencar Bezerra, G. N. Melo, 3DJPi: An Open-Source Web-Based 3D Simulator for Pololu's 3Pi Platform, in: Proceedings - 2019 21st Symposium on Virtual and Augmented Reality, SVR 2019, Institute of Electrical and Electronics Engineers Inc., 2019, pp. 52–58. URL: https://doi.org/10.1109/SVR.2019.00025.

[84] R. Marín, P. J. Sanz, The Human-Machine Interaction through the UJI Telerobotic Training

System, in: M. H. Hamza (Ed.), IASTED International Conference Robotics and Applications, RA 2003, June 25-27, 2003, Salzburg, Austria, IASTED/ACTA Press, 2003, pp. 47–52.

[85] H. S. Narman, C. Berry, A. Canfield, L. Carpenter, J. Giese, N. Loftus, I. Schrader, Augmented Reality for Teaching Data Structures in Computer Science, in: 2020 IEEE Global Humanitarian Technology Conference, GHTC 2020, Institute of Electrical and Electronics Engineers Inc., 2020, p. 9342932. URL: https://doi.org/10.1109/GHTC46280.2020.9342932.

[86] M. Nguyen, H. Le, P. M. Lai, W. Q. Yan, A web-based augmented reality platform using pictorial QR code for educational purposes and beyond, in: S. N. Spencer (Ed.), Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST, Association for Computing Machinery, 2019, p. 3364793. URL: https://doi.org/10.1145/3359996.3364793.

[87] V. T. Nguyen, K. Jung, S. Yoo, S. Kim, S. Park, M. Currie, Civil war battlefield experience: Historical event simulation using augmented reality technology, in: Proceedings - 2019 IEEE International Conference on Artificial Intelligence and Virtual Reality, AIVR 2019, Institute of Electrical and Electronics Engineers Inc., 2019, pp. 294–297. URL: https://doi.org/10.1109/AIVR46125.2019.00068.

[88] LATICE '14: Proceedings of the 2014 International Conference on Teaching and Learning in Computing and Engineering, IEEE Computer Society, USA, 2014. URL: https://www.computer.org/csdl/proceedings/latice/2014/12OmNrAdsty.

[89] Proceedings of 2015 International Conference on Interactive Mobile Communication Technologies and Learning, IMCL 2015, Institute of Electrical and Electronics Engineers Inc., 2015. URL: https://doi.org/10.1109/IMCL37494.2015.

[90] T. Tsiatsos, M. E. Auer (Eds.), 11th International Conference on Interactive Mobile Communication Technologies and Learning, IMCL2017, volume 725 of *Advances in Intelligent Systems and Computing*, Springer Verlag, 2018. URL: https://doi.org/10.1007/978-3-319-75175-7.

[91] Innovative Technologies and Learning: 4th International Conference, ICITL 2021, Virtual Event, November 29 – December 1, 2021, Proceedings, volume 13117 of *Lecture Notes in Computer Science*, Springer International Publishing, 2021. URL: http://doi.org/10.1007/978-3-030-91540-7. doi:10.1007/978-3-030-91540-7.

[92] N. Nordin, M. A. Markom, F. A. Suhaimi, S. Ishak, A web-based campus navigation system with mobile augmented reality intervention, Journal of Physics: Conference Series 1997 (2021) 012038. URL: https://doi.org/10.1088/1742-6596/1997/1/012038.

[93] S. L. Proskura, S. H. Lytvynova, The approaches to web-based education of computer science bachelors in higher education institutions, CTE Workshop Proceedings 7 (2020) 609–625. URL: https://doi.org/10.55056/cte.416.

[94] S. Proskura, S. Lytvynova, O. Kronda, N. Demeshkant, Mobile Learning Approach as a Supplementary Approach in the Organization of the Studying Process in Educational Institutions, in: O. Sokolov, G. Zholtkevych, V. Yakovyna, Y. Tarasich, V. Kharchenko, V. Kobets, O. Burov, S. Semerikov, H. Kravtsov (Eds.), Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume II: Workshops, Kharkiv, Ukraine, October 06-10, 2020, volume 2732 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 650–664. URL: https://ceur-ws.org/Vol-2732/20200650.pdf.

[95] G. Ryan, J. Murphy, M. Higgins, F. McAuliffe, E. Mangina, Work-in-Progress-Development of a Virtual Reality Learning Environment: VR Baby, in: D. Economou, A. Klippel, H. Dodds, A. Pena-Rios, M. J. W. Lee, D. Beck, J. Pirker, A. Dengel, T. M. Peres, J. Richter (Eds.), Proceedings of 6th International Conference of the Immersive Learning Research Network, iLRN 2020, Institute of Electrical and Electronics Engineers Inc., 2020, pp. 312–315. URL: https://doi.org/10.23919/iLRN47897.2020.9155203.

[96] S. Sendari, S. Wibawanto, J. Jasmine, M. Jiono, P. Puspitasari, M. Diantoro, H. Nur, Integrating Robo-PEM with AR Application for Introducing Fuel Cell Implementation, in: 7th International Conference on Electrical, Electronics and Information Engineering: Technological Breakthrough for Greater New Life, ICEEIE 2021, Institute of Electrical and Electronics Engineers Inc., 2021. URL: https://doi.org/10.1109/ICEEIE52663.2021.9616683.

[97] T. Sharkey, R. Twomey, A. Eguchi, M. Sweet, Y. C. Wu, Need Finding for an Embodied Cod-

ing Platform: Educators' Practices and Perspectives, in: M. Cukurova, N. Rummel, D. Gillet, B. McLaren, J. Uhomoibhi (Eds.), International Conference on Computer Supported Education, CSEDU - Proceedings, volume 1, Science and Technology Publications, Lda, 2022, pp. 216–227. URL: https://doi.org/10.5220/0011000200003182.

[98] N. Spasova, M. Ivanova, Towards augmented reality technology in CAD/CAM systems and engineering education, in: I. Roceanu (Ed.), eLearning and Software for Education Conference, National Defence University - Carol I Printing House, 2020, pp. 496–503. URL: https://doi.org/10.12753/2066-026X-20-151.

[99] D. Tennakoon, A. U. Usmani, M. Usman, A. Vasileiou, S. Latchaev, M. Baljko, U. T. Khan, M. A. Perras, M. Jadidi, TEaching Earth Systems Beyond the Classroom: Developing a Mixed Reality (XR) Sandbox, in: ASEE Annual Conference and Exposition, Conference Proceedings, American Society for Engineering Education, 2022.

[100] A. Toguchi, H. Sasaki, K. Mizuno, A. Shikoda, Build a prototype of new e-Learning contents by using the AR technology, in: IMSCI 2011 - 5th International Multi-Conference on Society, Cybernetics and Informatics, Proceedings, volume 1, International Institute of Informatics and Systemics, IIIS, 2011, pp. 261–264.

[101] A. Toguchi, H. Sasaki, K. Mizuno, A. Shikoda, Development of new e-Learning contents for improvement of laboratory courses by using the AR technology, in: IMSCI 2012 - 6th International Multi-Conference on Society, Cybernetics and Informatics, Proceedings, International Institute of Informatics and Systemics, IIIS, 2012, pp. 189–193.

[102] N. Tuli, A. Mantri, S. Sharma, Impact of augmented reality tabletop learning environment on learning and motivation of kindergarten kids, AIP Conference Proceedings 2357 (2022) 040017. URL: https://doi.org/10.1063/5.0080600.

[103] I. Wang, M. Nguyen, H. Le, W. Yan, S. Hooper, Enhancing Visualisation of Anatomical Presentation and Education Using Marker-based Augmented Reality Technology on Web-based Platform, in: Proceedings of AVSS 2018 - 2018 15th IEEE International Conference on Advanced Video and Signal-Based Surveillance, Institute of Electrical and Electronics Engineers Inc., 2019, p. 8639147. URL: https://doi.org/10.1109/AVSS.2018.8639147.

[104] S. Wongchiranuwat, C. Samat, Synthesis of theoretical framework for augmented reality learning environment to promote creative thinking on topic implementation of graphic design for grade 9 students, in: S. L. Wong, A. G. Barrera, H. Mitsuhara, G. Biswas, J. Jia, J.-C. Yang, M. P. Banawan, M. Demirbilek, M. Gaydos, C.-P. Lin, J. G. Shon, S. Iyer, A. Gulz, C. Holden, G. Kessler, M. M. T. Rodrigo, P. Sengupta, P. Taalas, W. Chen, S. Murthy, B. Kim, X. Ochoa, D. Sun, N. Baloian, T. Hoel, U. Hoppe, T.-C. Hsu, A. Kukulska-Hulme, H.-C. Chu, X. Gu, W. Chen, J. S. Huang, M.-F. Jan, L.-H. Wong, C. Yin (Eds.), ICCE 2016 - 24th International Conference on Computers in Education: Think Global Act Local - Main Conference Proceedings, Asia-Pacific Society for Computers in Education, 2016, pp. 639–641. URL: https://files.eric.ed.gov/fulltext/EJ1211500.pdf.

[105] ngrok, Unified Application Delivery Platform for Developers, 2024. URL: https://ngrok.com/.

[106] ngrok, Your Authtoken, 2024. URL: https://dashboard.ngrok.com/get-started/your-authtoken.

[107] TrackJS LLC, Remote JavaScript Debugger - RemoteJS, 2022. URL: https://remotejs.com/.

[108] MDN contributors, WebGL: 2D and 3D graphics for the web, 2023. URL: https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API.

[109] Three.js – JavaScript 3D Library, 2024. URL: https://threejs.org/.

[110] A. Klavins, 9 ideas for creating tech-infused augmented reality T-shirts, 2021. URL: https://overlyapp.com/blog/9-ideas-for-creating-tech-infused-augmented-reality-t-shirts/.

[111] Face Landmarks Detection, 2023. URL: https://github.com/tensorflow/tfjs-models/tree/master/face-landmarks-detection.

[112] Google LLC, Face landmark detection guide | MediaPipe | Google for Developers, 2023. URL: https://developers.google.com/mediapipe/solutions/vision/face_landmarker/.

[113] Face reference assets for Meta Spark Studio, 2023. URL: https://spark.meta.com/learn/articles/people-tracking/face-reference-assets.

[114] The face mask template in Adobe® Photoshop®, 2023. URL: https://spark.meta.com/learn/articles/

creating-and-prepping-assets/the-face-mask-template-in-Adobe.

[115] TensorFlow, 2024. URL: https://www.tensorflow.org/.

[116] TensorFlow.js models, 2024. URL: https://www.tensorflow.org/js/models.

[117] Hand Pose Detection, 2023. URL: https://github.com/tensorflow/tfjs-models/tree/master/hand-pose-detection.

[118] Find Pre-trained Models | Kaggle, 2024. URL: https://www.kaggle.com/models.

[119] Google, Teachable Machine, 2017. URL: https://teachablemachine.withgoogle.com/.

[120] MobileNet, 2023. URL: https://github.com/tensorflow/tfjs-models/tree/master/mobilenet.

[121] Speech Command Recognizer, 2024. URL: https://github.com/tensorflow/tfjs-models/tree/master/speech-commands.

[122] Pose Detection in the Browser: PoseNet Model, 2024. URL: https://github.com/tensorflow/tfjs-models/tree/master/posenet.