

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРИВОРІЗЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ
УНІВЕРСИТЕТ

Фізико-математичний факультет

Кафедра інформатики та прикладної математики

«Допущено до захисту»

В. о. завідувача кафедри

_____ Семеріков С. О.

Реєстраційний № _____

«_____» _____ 2022 р.

«_____» _____ 2022 р.

МЕТОДИКА НАВЧАННЯ РОЗРОБКИ
ДОПОВНЕНОЇ РЕАЛЬНОСТІ ДЛЯ WEB ІЗ
ІНТЕГРОВАНИМИ МОДЕЛЯМИ МАШИННОГО
НАВЧАННЯ

Кваліфікаційна робота студента групи Ім-17
ступінь вищої освіти «магістр»
спеціальності 014.09 Середня освіта (Інформатика)
Фокі Михайла Володимировича

Керівник: доктор педагогічних наук, професор,
старший дослідник
Семеріков Сергій Олексійович

Оцінка:

Національна шкала _____

Шкала ECTS _____ Кількість балів _____

Голова ЕК _____

Члени ЕК _____

ЗАПЕВНЕННЯ

Я, Фокі Михайло Володимирович, розумію і підтримую політику Криворізького державного педагогічного університету з академічної доброчесності. Запевняю, що ця кваліфікаційна робота виконана самостійно, не містить академічного плагіату, фабрикації, фальсифікації. Я не надавав і не одержував недозволену допомогу під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають покликання на відповідне джерело.

Із чинним Положенням про запобігання та виявлення академічного плагіату в роботах здобувачів вищої освіти Криворізького державного педагогічного університету ознайомлений. Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі порушення академічної доброчесності робота не допускається до захисту або оцінюється незадовільно.

(підпис)

ЗМІСТ

ВСТУП	4
1. ТЕОРІЯ ТА ПРАКТИКА НАВЧАННЯ РОЗРОБКИ ДОПОВНЕНОЇ РЕАЛЬНОСТІ ДЛЯ WEB	6
1.1. Бібліометричний аналіз джерел із освітніх застосувань WebAR	6
1.2. Засоби розробки доповненої реальності для Web	13
1.2.1. Налаштування веб-серверу та віддаленого налагоджувача	13
1.2.2. Застосування графічної бібліотеки для доповненої реальності у Web	20
1.2.3. Налаштування бібліотеки для доповненої реальності у Web	26
Висновки до розділу 1	35
2. ЕЛЕМЕНТИ МЕТОДИКИ НАВЧАННЯ РОЗРОБКИ ДОПОВНЕНОЇ РЕАЛЬНОСТІ ДЛЯ WEB ІЗ ІНТЕГРОВАНИМИ МОДЕЛЯМИ МАШИННОГО НАВЧАННЯ	38
2.1. Methodика розробки WebAR додатків для відстеження обличчя	38
2.1.1. Модель опорних точок обличчя	38
2.1.2. Накладання маски на обличчя	41
2.2. Methodика інтеграції моделей машинного навчання у WebAR додатки	44
2.2.1. Інтеграція стандартних моделей	44
2.2.2. Розробка користувацьких моделей	50
2.2.3. Інтеграція користувацьких моделей	62
Висновки до розділу 2	69
ВИСНОВКИ	72
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	76
ДОДАТКИ	91
А. Вихідний код користувацької бібліотеки loader.js	91

Б. Роздатковий матеріал “Класифікація риб”	93
В. Робочий аркуш “Дерево рішень”	96

ВСТУП

Актуальність теми. Доповнена реальність у Web (Web-based Augmented Reality, або WebAR) є одним із найпоширеніших способів об'єднання реального та віртуального на мобільних Інтернет-пристроях [30; 120]. Розробка веб-додатків доповненої реальності відрізняється від інших способів розробки тим, що є крос-платформовою і не вимагає встановлення розроблених додатків, що суттєво підвищує рівень мобільності програмного забезпечення порівняно із традиційними мобільними додатками [16].

Наразі найвідоміша у світі некомерційна бібліотека для розробки WebAR – AR.js [12], започаткована Жеромом Етьєнном (так, у [92] подано системний опис можливостей застосування AR.js для формування професійних компетентностей майбутніх учителів STEM-дисциплін), проте ХіуКім Юен (HiuKim Yuen [126]), один із розробників AR.js, створив нову бібліотеку під назвою MindAR [127] – більш компактну та технологічно розвинену, але, на відміну від AR.js, маловідому.

AR.js та MindAR побудовані на класичних рушіях ARToolKit та OpenCV відповідно, що наразі є промисловим стандартом. Водночас, якщо AR.js зорієнтована на опрацювання насамперед простих маркерів розміром до 16×16 , то MindAR – на природні зображення складної структури. Інша особливість MindAR, що робить її доцільним засобом навчання – включення до її складу відомої бібліотеки машинного навчання TensorFlow [104], що надає потенційні можливості для інтеграції моделей машинного навчання у WebAR додатки з метою створення високоінтерактивних і цікавих ефектів, наприклад, використання жестів рук або міміки для управління AR-контентом.

Саме на реалізацію освітньої складової цього потенціалу й спрямоване дослідження на тему: **“Методика навчання розробки доповненої реальності для Web із інтегрованими моделями машинного навчання”**.

Об'єкт дослідження – процес навчання розробки імерсивних освітніх ресурсів.

Предмет дослідження – методика навчання розробки WebAR додатків із інтегрованими моделями машинного навчання.

Мета дослідження – розробити окремі елементи методики навча-

ння розробки доповненої реальності для Web із інтегрованими моделями машинного навчання.

Відповідно до мети визначено такі основні **завдання дослідження**:

1. Виконати бібліометричний аналіз джерел із освітніх застосувань WebAR.
2. Дібрати засоби розробки доповненої реальності для Web.
3. Розробити та апробувати методику розробки WebAR додатків для відстеження обличчя.
4. Розробити та апробувати методику інтеграції моделей машинного навчання у WebAR додатки.

Методи дослідження: аналіз джерел, методичне проектування, методи інженерії програмного забезпечення.

Практичне значення результатів дослідження полягає у розробці окремих елементів методики навчання розробки доповненої реальності для Web із інтегрованими моделями машинного навчання.

Структура роботи. Робота складається зі вступу, двох розділів, висновків до них, загальних висновків, списку використаних джерел (127 найменувань, серед яких 125 – англійською), 3 додатків (на 6 сторінках). Загальний обсяг роботи – 96 сторінок.

РОЗДІЛ 1

ТЕОРІЯ ТА ПРАКТИКА НАВЧАННЯ РОЗРОБКИ ДОПОВНЕНОЇ РЕАЛЬНОСТІ ДЛЯ WEB

1.1. Бібліометричний аналіз джерел із освітніх застосувань WebAR

Для виконання систематизованого бібліометричного аналізу за запитом “WebAR” та “Web-based augmented reality for education” був застосований VOSviewer версії 1.6.18 [21].

В якості джерела даних для першого запиту був обраний Crossref із пошуком по заголовках документів, що надало можливість відібрати 19 документів 2017–2022 рр. Відібрані документи були проаналізовані за кількістю разів, які вони цитуються разом із іншими документами.

Із 92 джерел, що цитувались у 19 документах, 26 цитуються спільно більше одного разу, утворюючи лише 1 кластер (рис. 1.1), що складають роботи «Development of career guidance quests using WebAR» [30], «WebAR development tools: comparative analysis» [121] та «WebAR development tools: An overview» [30], виконані під керівництвом С. О. Семерікова.

В якості джерела даних для другого запиту був обраний Scopus із пошуком по заголовках, анотаціях та ключових словах документів, що надало можливість відібрати 93 документи 2001–2023 рр. (рис. 1.2), 66 з яких припадають на останні 5 років. Більшу частину з них складають статті у журналах (58 [1–3; 8; 9; 13; 14; 17; 18; 24–27; 29; 32–37; 39–43; 45; 49; 53; 55–58; 60; 63; 65; 67–69; 71; 76; 78–80; 82; 86; 87; 89; 91; 93; 95; 99; 106; 112; 114; 117; 118; 122; 125]), меншу – книги (4 [50; 54; 59; 64]) та статті у матеріалах конференцій (31 [4–7; 10; 11; 15; 20; 23; 28; 31; 38; 46; 47; 52; 62; 63; 70; 73; 75; 82–85; 96; 97; 101; 111; 116; 123; 124]).

Із 301 автора 93 документів 2 та більше разів цитувались 27 авторів, а 3 та більше – 9. На рис. 1.3 показано семантичну мережу ключових слів у документах за запитом “Web-based augmented reality for education”. Розподіл ключових слів за кластерами (рис. 1.4) подано у табл. 1.1.

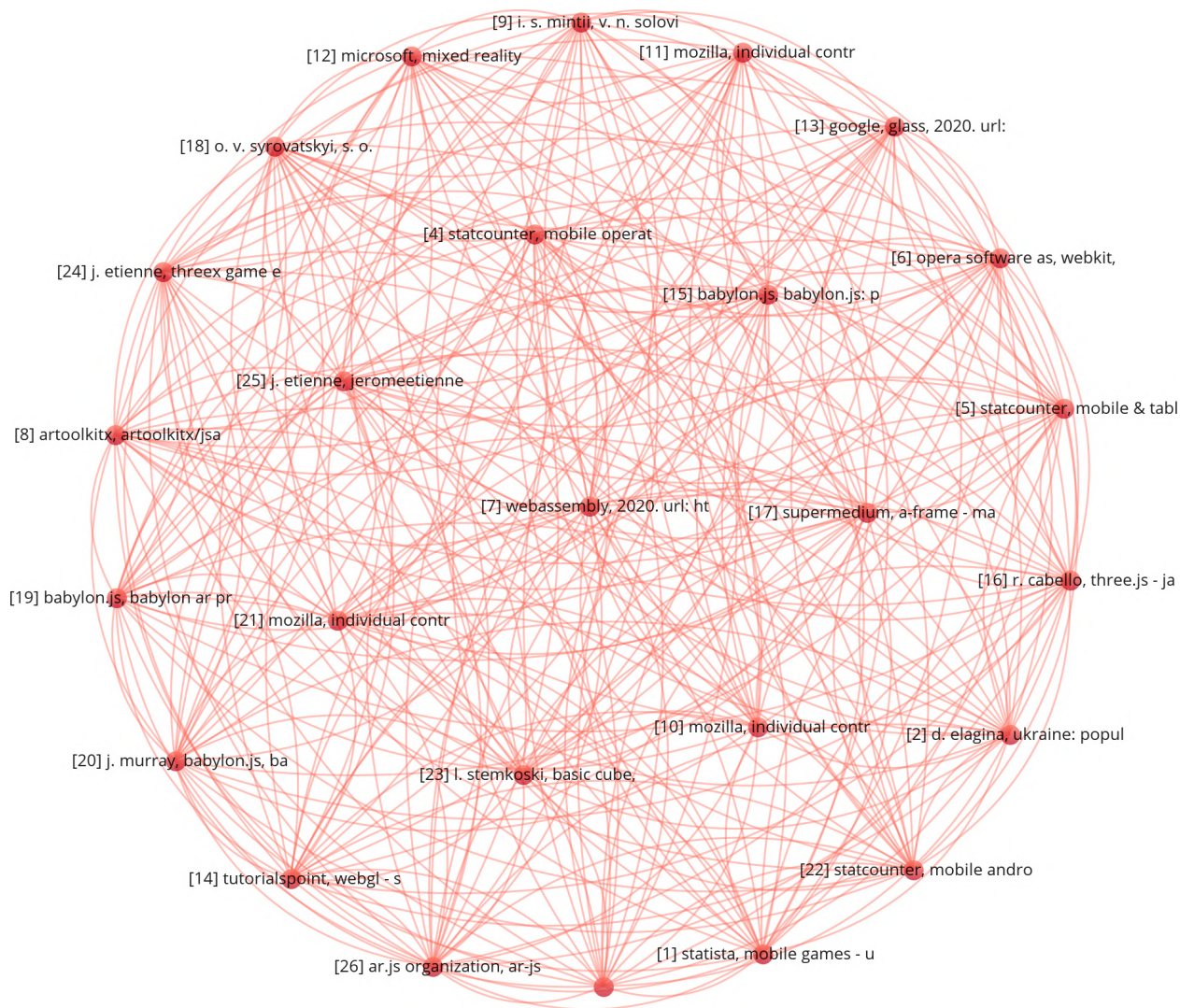


Рис. 1.1. Семантична мережа посилань у документах за запитом “WebAR”.

Перший кластер (виділений червоним кольором на рис. 1.4 та у табл. 1.1) пов’язує **базові концепти доповненої реальності в освіті**: доповнену та віртуальну реальність із освітою (зокрема, медичною) та навчанням людей, зокрема, із використанням смартфонів.

Доповнена реальність виступає системотвірним елементом – вона пов’язує усі кластери та сама є пов’язаною з усіма іншими концептами.

У аналізованих документах *віртуальна реальність* не пов’язується із традиційним викладанням, мобільним навчанням та інтернет/веб-орієнтованим навчанням. Тут слід відрізнити віртуальну реальність від віртуальних навчальних середовищ, які якраз охоплюють перелічені поняття.

Концепт *освіти* також відноситься до майже універсальних – він не пов’язується лише з інтерфейсами користувача та AR-додатками.

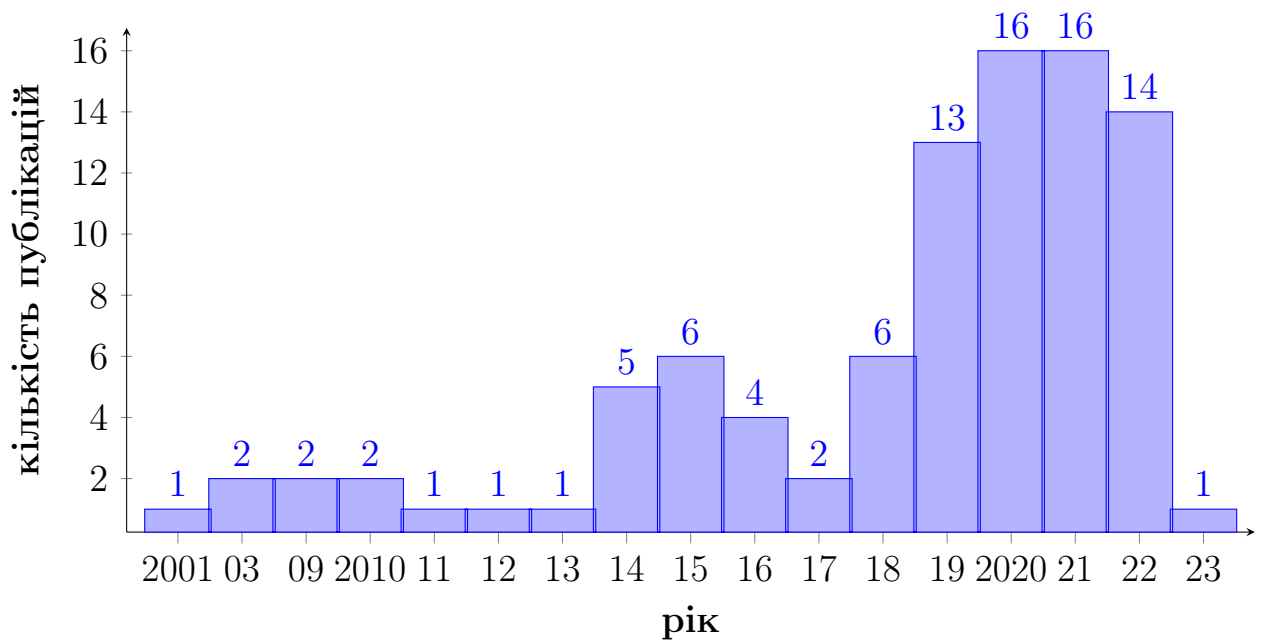


Рис. 1.2. Розподіл документів за роками (запит “Web-based augmented reality for education”).

Досить показові зв’язки *медичної освіти* з іншими кластерами: у другому кластері – із поняттями *курукулуму*, *ІКТ навчання*, *комп’ютерно зорієнтованого викладання*, *електронного навчання* та *студентами*, у третьому – із *веб-сайтами* та *педагогічними технологіями доповненої реальності*, у четвертому – із *дистанційною освітою*.

Навчання (у сенсі *учіння*) пов’язане у другому кластері з *викладанням*, *студентами*, *ІКТ навчання*, *комп’ютерно зорієтованим викладанням*, *електронним навчанням* та *інтерфейсами користувача*, у третьому – із *веб-сайтами* та *мотивацією*. Дане поняття не має прямих зв’язків із *дистанційною освітою*.

Концепт *людини* поза межами свого кластеру пов’язаний зі *студентами* та *електронним навчанням* у другому кластері та *веб-сайтами* у третьому.

Поза межами свого кластеру, *інтернет/веб-орієнтоване навчання* пов’язане лише з *традиційним викладанням* у другому кластері.

Нарешті, *смартфони* пов’язані у другому кластері з *викладанням*, *студентами*, *ІКТ навчання*, *електронним навчанням* та *інженерною освітою*, у третьому – із *веб-сайтами* та *AR-додатками*.

Другий кластер (виділений зеленим кольором на рис. 1.4 та у

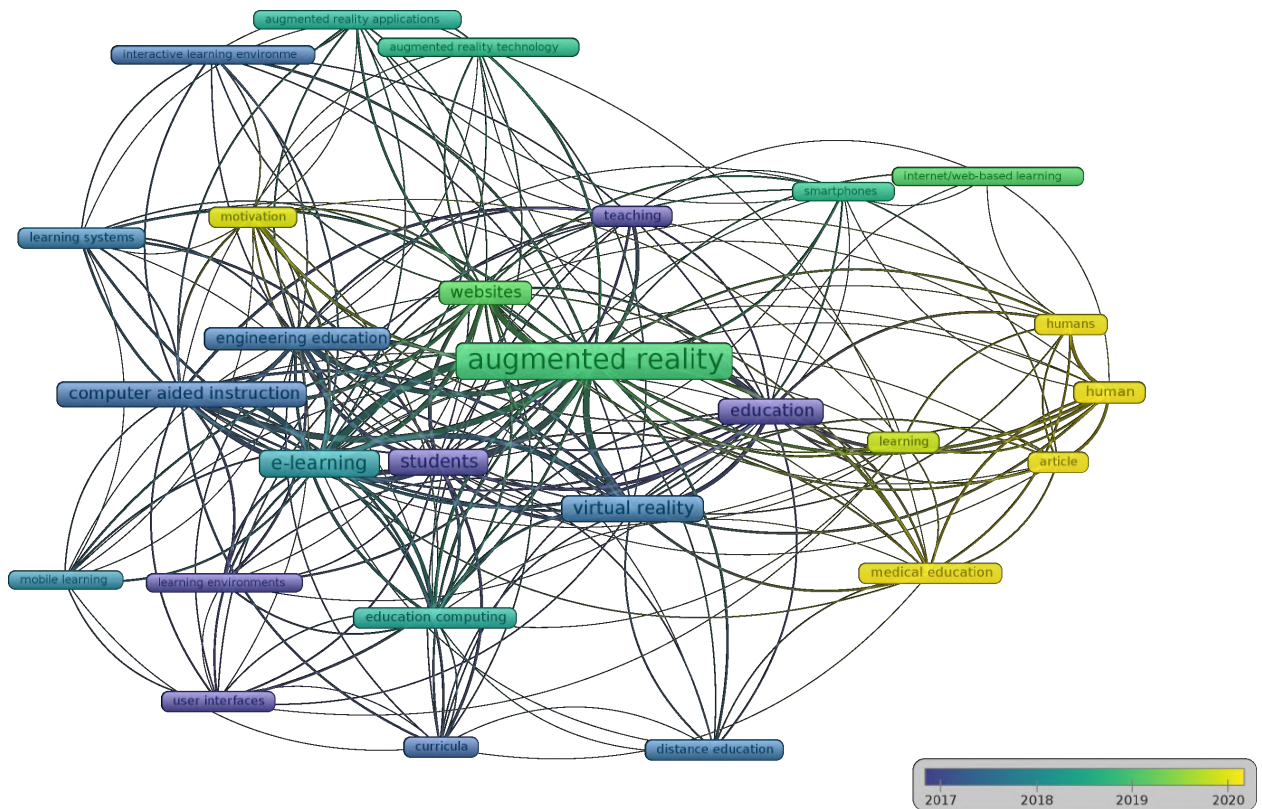


Рис. 1.3. Семантична мережа ключових слів у документах за запитом “Web-based augmented reality for education”.

табл. 1.1) пов’язує **концепти проектування навчального середовища**: викладання, інженерну освіту, комп’ютерно зорієнтоване викладання, електронне навчання, студентів, мобільне навчання, навчальні середовища, ІКТ навчання та курикулуми.

Центральними у другому кластері є концепти “електронне навчання” та “студенти”, що також відноситься до майже універсальних – формально, вони не пов’язується лише з інтернет/веб-орієнтованим навчанням через його синонімічність із електронним навчанням.

Комп’ютерно зорієнтоване викладання пов’язане із концептами першого (доповнена та віртуальна реальність, освіта (включно із медичною) та навчання) та третього (мотивація, веб-сайти, системи навчання, інтерактивні навчальні середовища, AR-додатки, технології доповненої реальності) кластерів.

Концепт *викладання* пов’язаний у першому кластері із доповненою реальністю, освітою та навчанням, смартфонами та інтернет/веб-орієнтованим навчанням, а у третьому – із веб-сайтами, AR-додатками та технологіями доповненої реальності.

Таблиця 1.1

Розподіл ключових слів за кластерами (документи за запитом “Web-based augmented reality for education”).

Кластер 1	Кластер 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
Кластер 3	Кластер 4
augmented reality applications	distance education
augmented reality technology	
interactive learning environments	
learning systems	
motivation	
websites	

Інженерна освіта пов’язана у першому кластері із доповненою та віртуальною реальністю, освітою та смартфонами, та з усіма концептами третього і четвертого кластерів.

ІКТ навчання пов’язані у першому кластері із доповненою та віртуальною реальністю, освітою (включно із медичною) та навчанням, смартфонами, у третьому – із мотивацією, системами навчання та веб-сайтами, у четвертому – із дистанційною освітою.

Поза межами свого кластеру, *навчальні середовища* пов’язані лише з освітою, доповненою та віртуальною реальністю з першого кластеру й веб-сайтами із третього.

Аналогічно, *мобільне навчання* пов’язане із освітою та доповненою реальністю із першого кластеру й мотивацією, веб-сайтами та системами

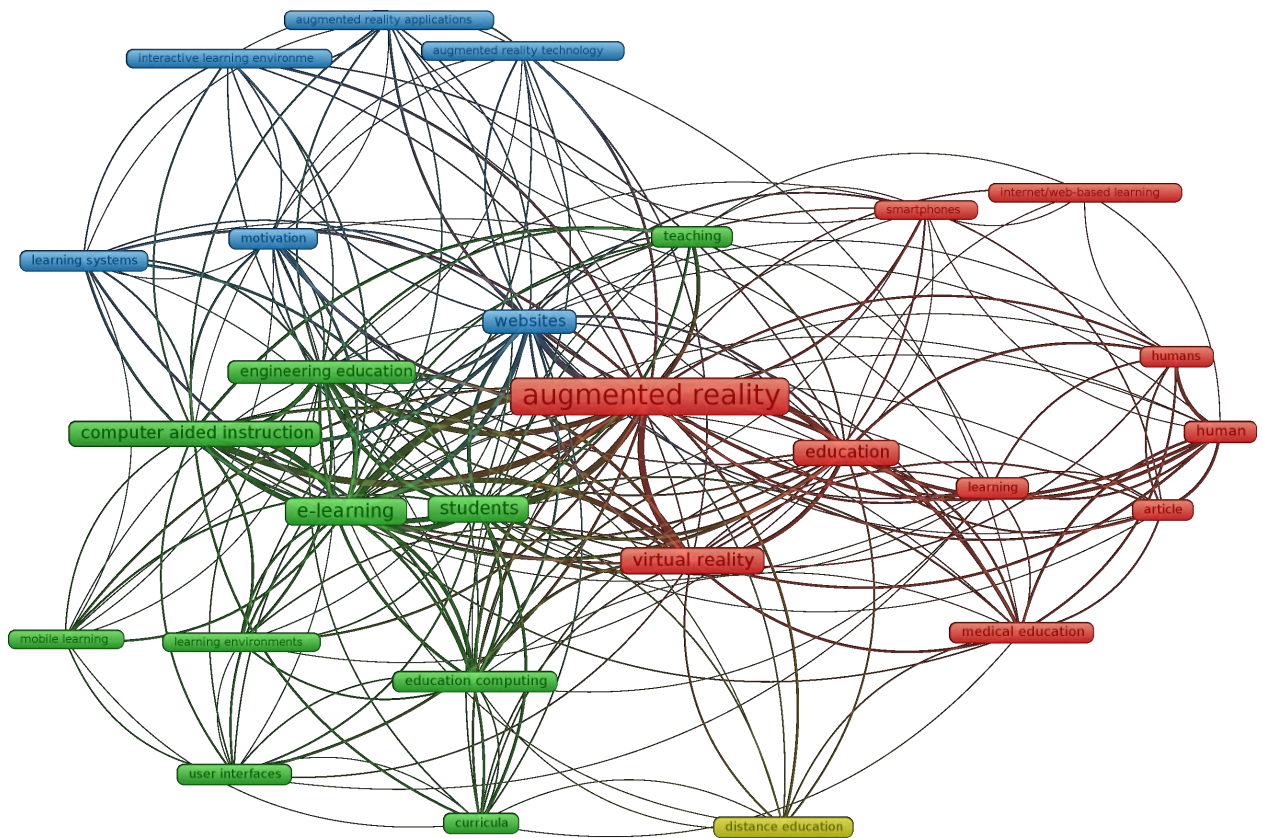


Рис. 1.4. Розподіл ключових слів за кластерами.

навчання із третього.

Інтерфейси користувача мають зв'язки із концептами першого (навчання, доповнена та віртуальна реальність) та третього (мотивація, веб-сайти) кластерів.

Курикулами пов'язані у першому кластері із освітою (включно з медичною), доповненою та віртуальною реальністю, у третьому – із веб-сайтами та у четвертому із дистанційною освітою.

Третій кластер (виділений блакитним кольором на рис. 1.4 та у табл. 1.1) пов'язує **концепти реалізації імерсивного навчального середовища**: веб-сайти, мотивацію, системи навчання, інтерактивні навчальні середовища, AR-додатки та технології доповненої реальності.

Центральними у третьому кластері є *веб-сайти*, що відноситься до майже універсальних концептів – формально, вони не пов'язується лише з інтернет/веб-орієнтованим навчанням через перекриття відповідних понять.

Концепт *мотивації* пов'язаний у першому кластері із доповненою та віртуальною реальністю, освітою та навчанням, а у третьому – із еле-

ектронним та мобільним навчанням, ІКТ навчання, інтерфейсами користувача, комп'ютерно зорієнтованим викладанням, студентами та інженерною освітою.

Системи навчання пов'язані у першому кластері із доповненою та віртуальною реальністю і освітою, а у третьому – із електронним та мобільним навчанням, ІКТ навчання, комп'ютерно зорієнтованим викладанням, студентами та інженерною освітою.

Схожі зв'язки мають й *інтерактивні навчальні середовища*: у першому кластері – із доповненою та віртуальною реальністю і освітою, а у третьому – із електронним навчанням, комп'ютерно зорієнтованим викладанням, студентами та інженерною освітою.

Природно, *AR-додатки* пов'язані у першому кластері з доповненою реальністю та смартфонами, а у другому – із електронним навчанням, комп'ютерно зорієнтованим викладанням, студентами та інженерною освітою.

Технології доповненої реальності пов'язані у першому кластері з доповненою та віртуальною реальністю і освітою (включно з медичною), а у другому – із електронним навчанням, комп'ютерно зорієнтованим викладанням, студентами та інженерною освітою.

Четвертий кластер (виділений жовтим кольором на рис. 1.4 та у табл. 1.1) містить **концепт дистанційної освіти**, пов'язаний у першому кластері із концептами доповненої та віртуальної реальності і концептом освіти (включно з медичною), у другому – із концептами студента, інженерної освіти, ІКТ навчання, електронного навчання та курикулуму, та у третьому – із концептом веб-сайту.

Аналіз розподілу концептів за густиною зв'язків (рис. 1.5) та часом надає можливість визначити, що найбільш давні (до 2015 року) дослідження були зосереджені на питаннях інтерфейсів користувача та їх застосування в освіті. У роботах 2016 року центр уваги зміщувався у напрямі дослідження впливу на студентів викладання у навчальних середовищах. Дослідження 2017 року актуалізували концепти віртуальної реальності, інтерактивних навчальних середовищ, курикулуму, комп'ютерно зорієнтованого навчання, зокрема, у інженерній освіті. Центром уваги досліджень 2018 року стали ІКТ навчання, застосування смартфонів, AR-

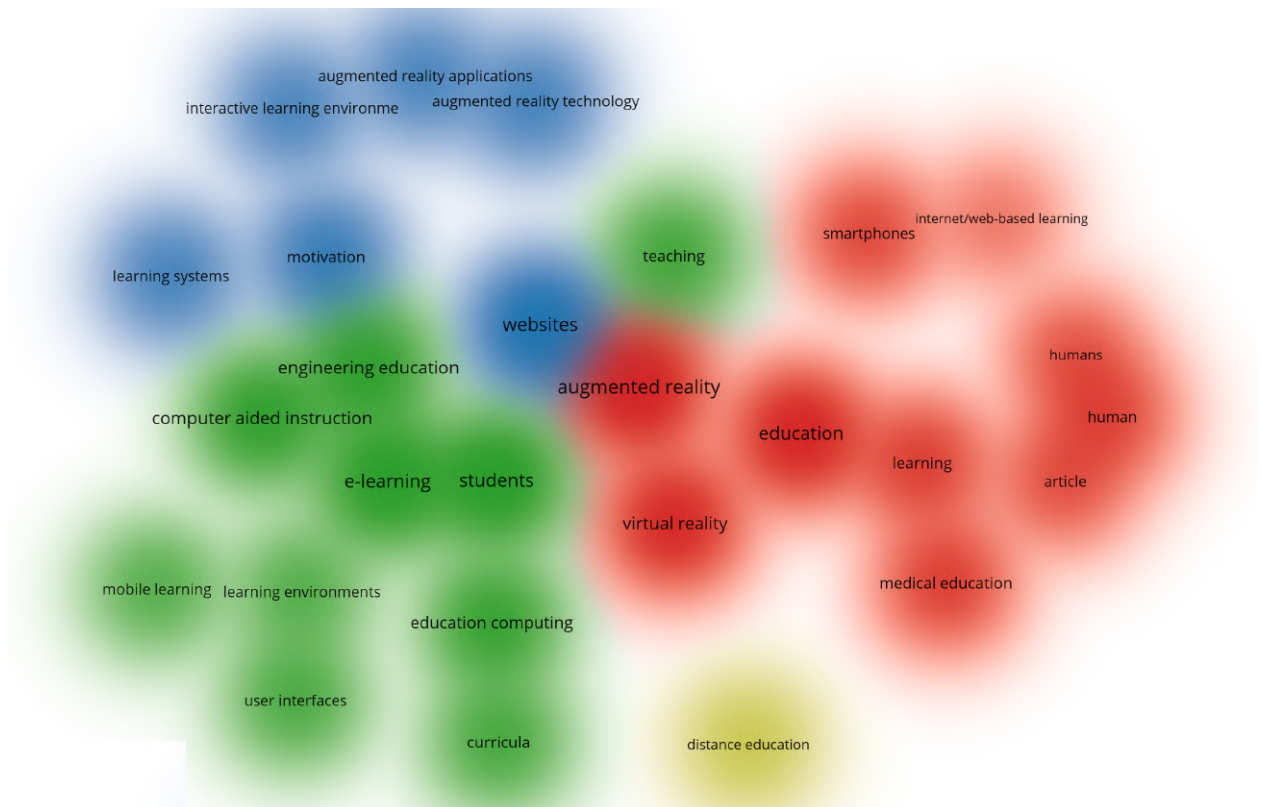


Рис. 1.5. Густина зв'язків ключових слів за запитом “Web-based augmented reality for education”.

додатків та педагогічних технологій доповненої реальності.

WebAR стає предметом уваги досліджень 2019 року – у них підіймаються питання застосування смартфонів, інтернет/веб-орієнтованого навчання та доповненої реальності. У 2020 року вплив пандемії COVID-19 додав питання мотивації навчання та медичної освіти. Новий елемент останніх за часом досліджень – це доповнення (аугментація) людини.

1.2. Засоби розробки доповненої реальності для Web

1.2.1. Налаштування веб-серверу та віддаленого налагоджувача

Для розробки мовами HTML та JavaScript основними засобами розробки є простий текстовий редактор і веб-браузер (рис. 1.6), в якому можна відкрити звичайну веб-сторінку HTML, збережену локально.

Однак це може не спрацювати для додатків, які потребують використання камери. Крім того, може знадобитися час від часу тестувати додатки на власних мобільних пристроях, тому краще встановити локальний

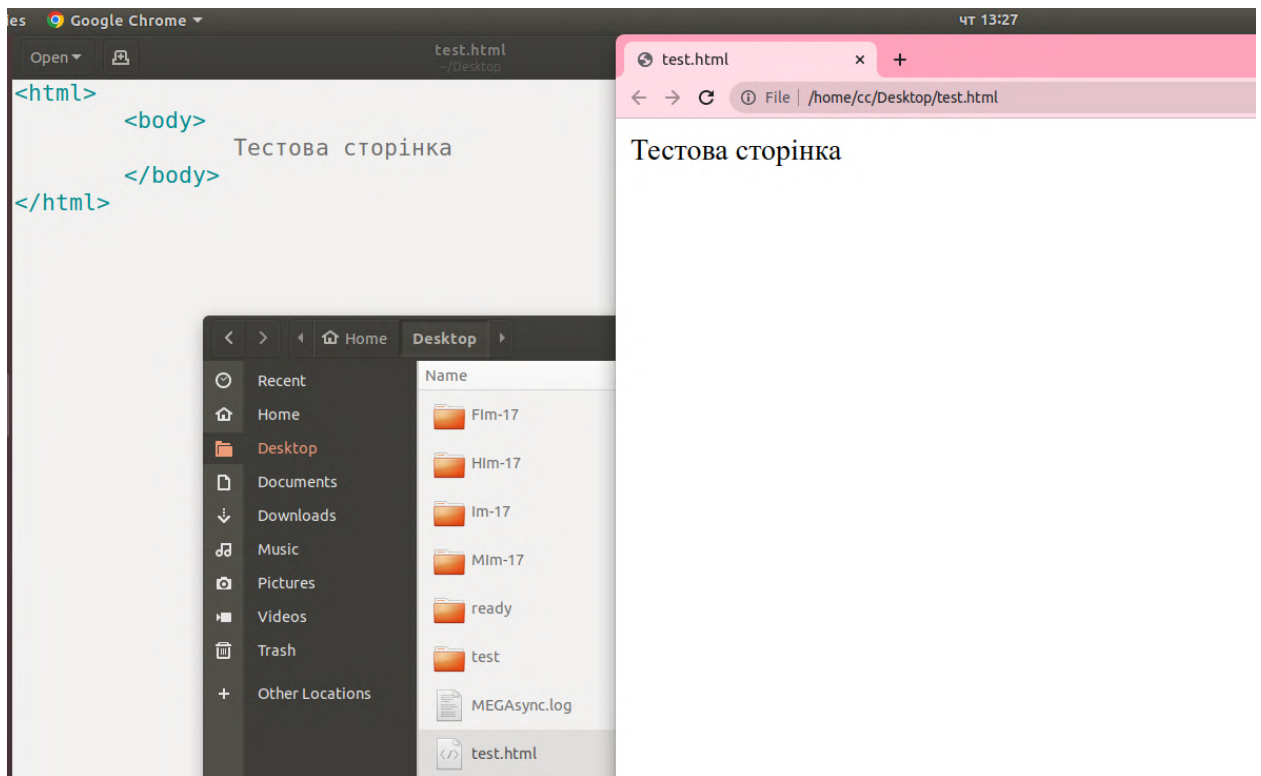


Рис. 1.6. Текстовий редактор – основний засіб розробки.

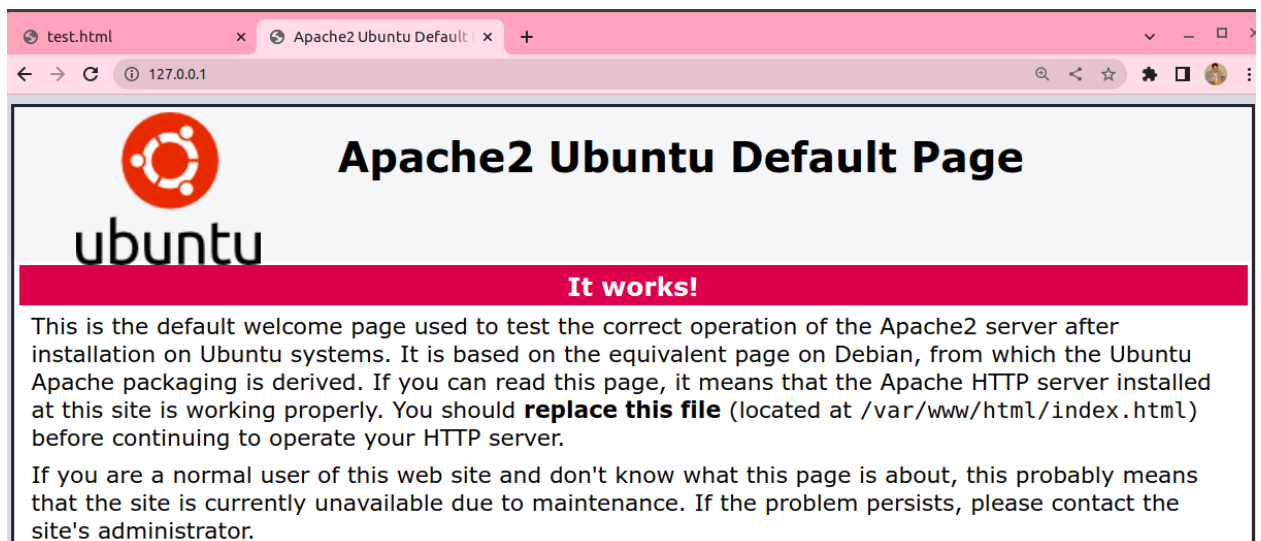


Рис. 1.7. Робота локального веб-серверу Apache.

веб-сервер (рис. 1.7).

Альтернативний спосіб – встановити розширення для Chrome, яке називається “Web Server for Chrome” [119] та є дуже простим у використанні. Після встановлення необхідно запустити сервер, а потім обрати кореневий каталог для веб-сторінок та обрати можливість доступу у локальній мережі, щоб інші пристрої також могли отримати доступ до веб-сторінки (рис. 1.8, 1.9); по завершенні роботи веб-сервер можна зупинити.

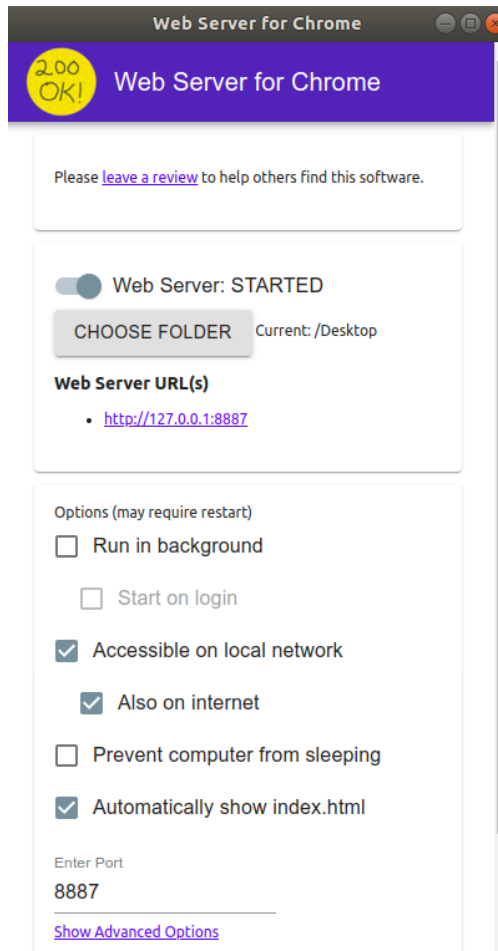


Рис. 1.8. Налаштування веб-серверу для Chrome.

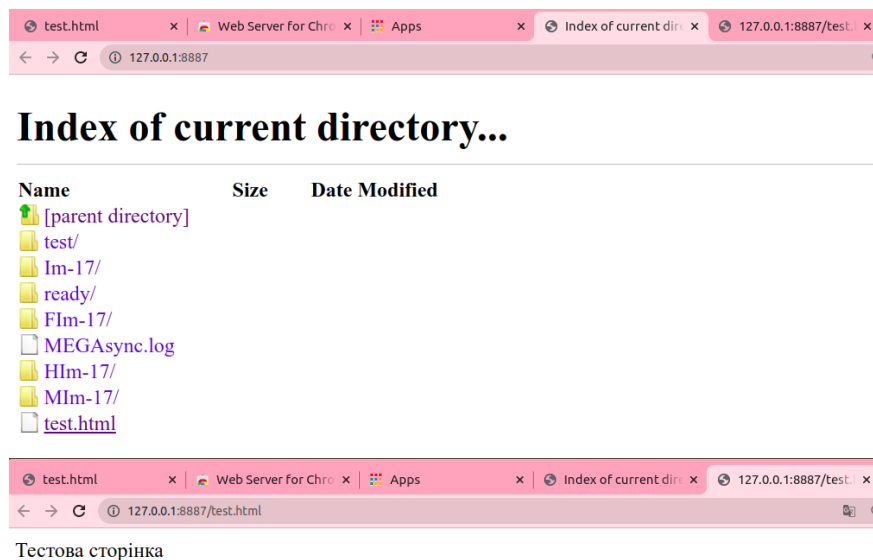


Рис. 1.9. Робота веб-серверу для Chrome.

Корисним може бути вибір протоколу HTTPS у додаткових налаштуваннях – без його використання мобільний пристрій може не надати доступ до камери.



Рис. 1.10. Доступ до веб-сторінки у локальній мережі з різних пристроїв.

Технічно можна виконувати всю роботу з розробки та тестування безпосередньо на настільному браузері, але іноді все ж таки доцільно спробувати на мобільному телефоні (рис. 1.10).

Якщо пристрої підключені до однієї локальної мережі, у якій немає брандмауера, проблем із доступом до веб-серверу немає. Однак, якщо точка доступу до мережі знаходиться за брандмауером, можна використовувати

ngrok [77] для того, щоб виконати перенаправлення трафіку з порту, доступ до якого обмежений.

Після встановлення ngrok та створення облікового запису на сайті [77] необхідно зареєструвати агент ngrok [94] та запустити його, вказавши в якості параметру протокол (наприклад, http) та номер порту, доступ до якого закриває брандмауера (наприклад, 8887) (рис. 1.11).

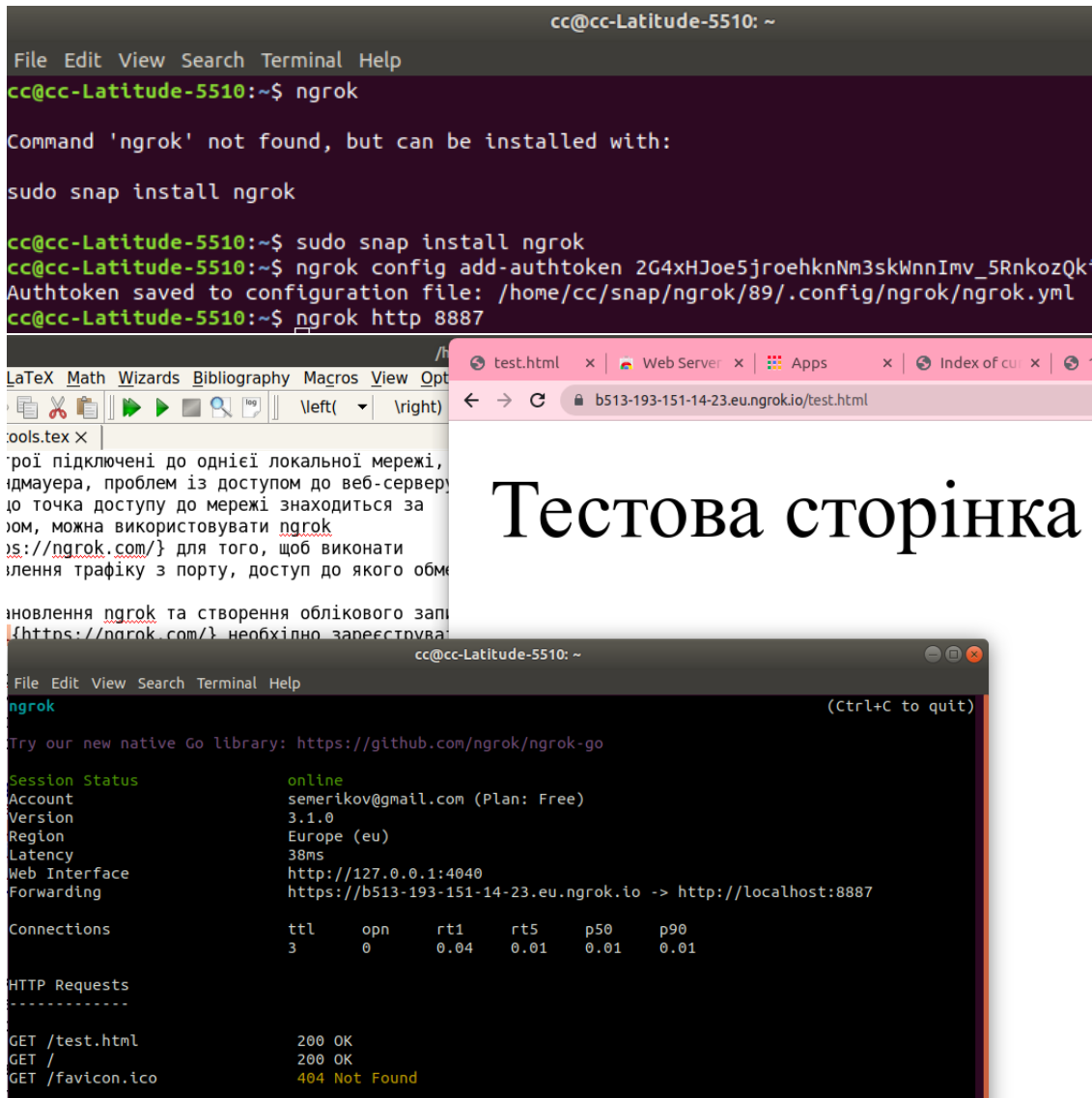


Рис. 1.11. Застосування ngrok для обходу брандмауера.

Після запуску ngrok надає глобальне Інтернет-посилання за протоколом HTTPS – але лише у той час, коли працюють одночасно локальний веб-сервер та перенаправлення ngrok.

Традиційно, налагодження веб-додатків передбачає перегляд консолі веб-браузера, куди виводяться повідомлення, що стосуються налагодження програми (рис. 1.12):

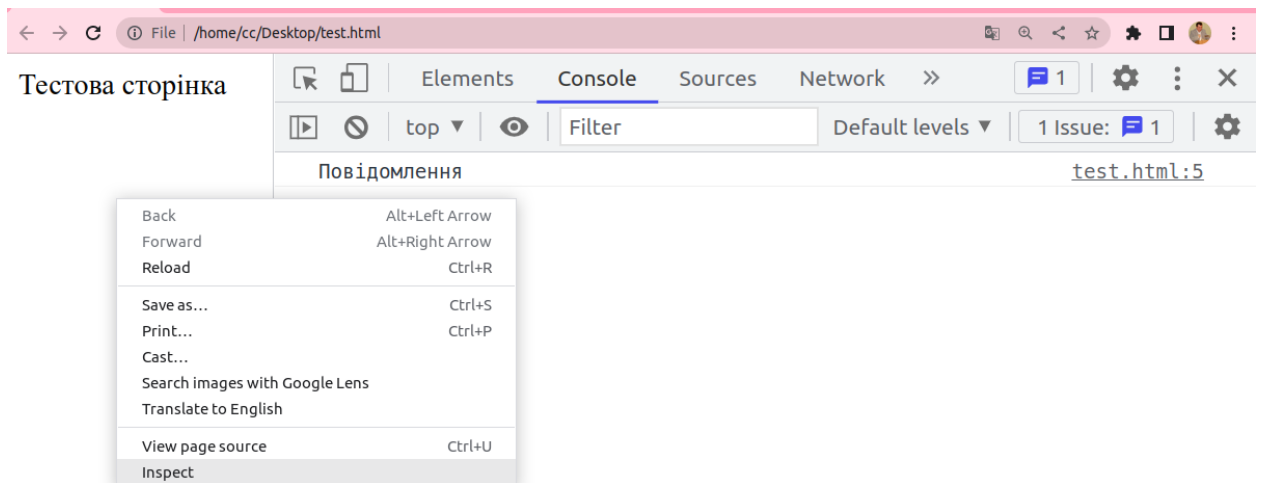


Рис. 1.12. Inspect – перегляд консолі веб-браузера.

```
<html>
  <body>
    Тестова сторінка
    <script>
      console.log("Повідомлення");
    </script>
  </body>
</html>
```

Однак на мобільному пристрої це може бути не так просто. Тут допоможе RemoteJS [110] – натиснувши після переходу на сайт кнопку “Start Debugging”, отримаємо код агенту RemoteJS виду

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

Цей код необхідно скопіювати і вставити безпосередньо у веб-сторінку:

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

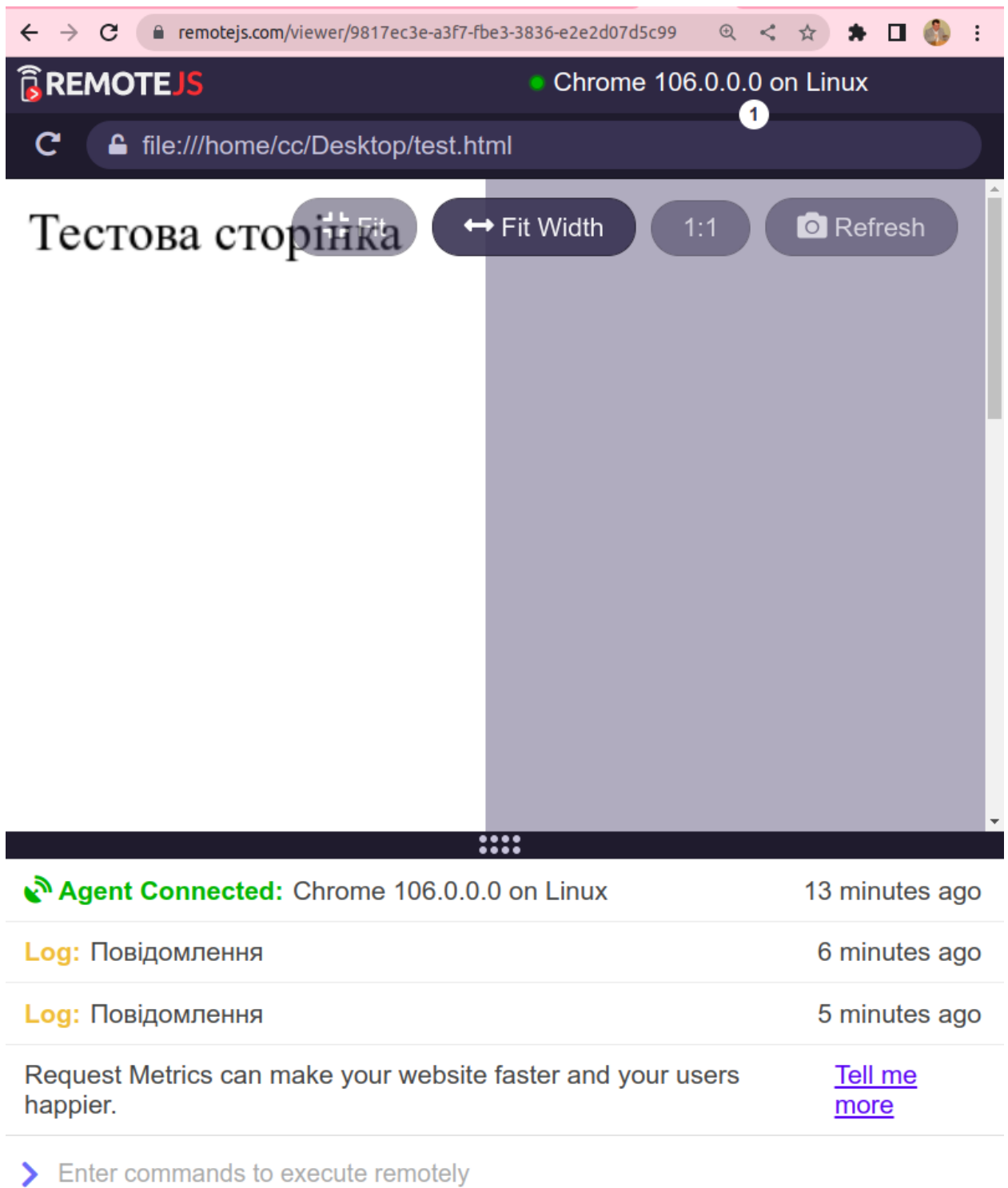


Рис. 1.13. Віддалене налагодження за допомогою агенту RemoteJS.

```
</script>
</head>
<body>
  Тестова сторінка
  <script>
```

```
        console.log("Повідомлення");  
    </script>  
</body>  
</html>
```

Після цього всі налагоджувальні повідомлення будуть надіслані на веб-сторінку з адресою https://remotejs.com/viewer/agent_code, де `agent_code` – значення змінної `data-consolejs-channel` (рис. 1.13).

1.2.2. Застосування графічної бібліотеки для доповненої реальності у Web

WebGL [72] – JavaScript API для рендеринга 3D-графіки в браузерах. Він є крос-платформним стандартом відображення, який підтримують всі основні браузери. Проте низькорівневий код WebGL складно читати та писати, тому були створені більш зручні для користувача бібліотеки.

Three.js [108] є однією з таких бібліотек. Її автор Рікардо Мігель Кабелло, також відомий як `mrdob`, є одним із піонерів використання WebGL, тому ця бібліотека часто використовується при побудові інших бібліотек. Більшість WebAR SDK підтримують Three.js, тому вона дійсно потребує опанування для ефективної розробки веб-додатків з доповненою реальністю.

Щоб зрозуміти, як на високому рівні працює Three.js, доцільно провести аналогію з роботою фото- чи кінорежисера, який:

- 1) налаштовує сцену шляхом розташування на ній об'єктів;
- 2) рухає камеру, щоб зафіксувати кадри з різних позицій та ракурсів.

Three.js не є спеціалізованою бібліотекою для доповненої реальності – вона містить суттєво більше функціональності, в тому числі тієї, що є більш придатною для веб-VR (освітлення, камери та ін.) (рис. 1.14).

Для створення додатку із застосуванням Three.js необхідно створити сторінку HTML, особливістю якою буде відсутність тіла, адже для генерації її вмісту буде використовуватись код JavaScript, що зазвичай розташовується між тегамі `<script>` та `</script>`. Проте це найкраща практика, тому доцільним є розміщення коду в окремому файлі `main.js`:

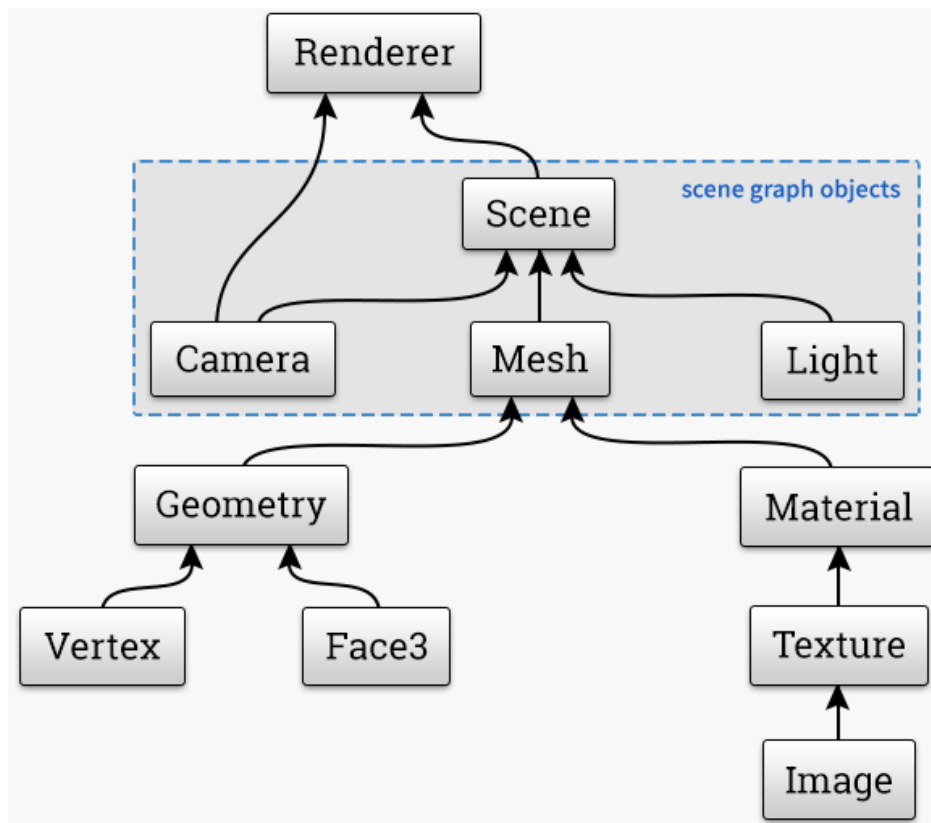


Рис. 1.14. Загальна структура Three.js.

```

<html>
  <head>
    <script src="./main.js" type="module"></script>
  </head>
  <body>
  </body>
</html>

```

Перше, що необхідно зробити, це імпортувати 3D бібліотеку. Наразі багато бібліотек, розміщених у хмарі, можна імпортувати безпосередньо через механізм CDN (content delivery network – мережу доставляння контенту), таких, як unpkg [113]. На сторінці документації Three.js [61] подано відповідні приклади того, як це робити.

Так, при використанні звернення до CDN за шляхом `https://unpkg.com/three/` буде отриманий повний перелік версій Three.js, доступних через CDN. У випадку, коли необхідно зафіксувати номер версії бібліотеки (адже її функціональність змінюється – не лише з’являються нові можливості, а й зникають застарілі), доцільно чітко його вказати: наприклад,

замість узагальненого `https://unpkg.com/three/build/three.module.js` вказати `https://unpkg.com/three@0.145.0/build/three.module.js`.

Імпорт із CDN

```
import * as THREE from "https://unpkg.com/three/build/three.module.js"
```

чи локального файлу

```
import * as THREE from "../three.js-r145/build/three.module.js";
```

вимагає додавання атрибуту `module` до тегу `script`:

```
<script src="./main.js" type="module"></script>
```

Намагання переглянути цей документ HTML локально у більшості сучасних браузерів завершиться невдачею через заборону локального (за протоколом `file://`) завантаження модулів, тому необхідним є застосування локального чи віддаленого веб-серверу.

Зазвичай код JavaScript має виконуватись після завершення завантаження документа HTML. Для того, щоб убезпечити це, слід додати обробник події `DOMContentLoaded`, усередині якого розташувати код:

```
document.addEventListener("DOMContentLoaded", () => {  
  const scene = new THREE.Scene();  
  
  const geometry = new THREE.BoxGeometry(1, 1, 1);  
  const material = new THREE.MeshBasicMaterial({color: "#0000FF"});  
  const cube = new THREE.Mesh(geometry, material);  
  cube.position.set(0, 0, -2);  
  cube.rotation.set(0, Math.PI/4, 0);  
  scene.add(cube);  
  
  const camera = new THREE.PerspectiveCamera();  
  camera.position.set(1, 1, 5);  
  
  const renderer = new THREE.WebGLRenderer({alpha: true});  
  renderer.setSize(500, 500);
```

```
renderer.render(scene, camera);

document.body.appendChild(renderer.domElement);
});
```

Як показано на рис. 1.14, основою є сцена – для її створення необхідно викликати конструктор без параметрів класу **Scene** (змінна `scene` – об'єкту класу **Scene** створюється динамічно за допомогою виклику `new`).

Створення об'єктів у Three.js відбувається у три кроки:

- 1) визначення геометрії об'єкту – векторів позиції, кольорів та ін.: так, **BoxGeometry** відповідає за прямокутний паралелепіпед;
- 2) визначення матеріалу – способу рендерингу об'єкту (його оптичні властивості – колір, фактура, блиск тощо): так, **MeshBasicMaterial** відповідає матеріалу, що має власний колір і не відбиває промені;
- 3) композиція геометрії та матеріалу виконується за допомогою **Mesh**.

Створений у такий спосіб куб буде розташований у початку координат. Для зміни його позиції скористаємось властивістю `position`, успадкованою класом **Mesh** від свого батька – **Object3D**. Дана властивість є об'єктом класу **Vector3**, а `set` – його методом.

Аналогічно, властивість `rotation` зберігає кути нахилу об'єкту в радіанах. До речі, документація Three.js (див., наприклад, [19]) містить інтерактивні демонстрації, що надають можливість переглянути різні об'єкти та модифікувати їх параметри. Рікардо Кабелло надає можливість конструювання сцени за допомогою візуального редактора [109] – це може суттєво прискорити та полегшити процес її створення.

Всі об'єкти розміщуються на сцені за допомогою методу `add`.

Наступний об'єкт, що створюється – перспективна камера (**PerspectiveCamera**). Параметрами конструктора **PerspectiveCamera** є складові зрізаної піраміди огляду: перший – вертикальне поле зору (кут у градусах), другий – співвідношення сторін камери, третій – найближча площина, четвертий – найдальша (рис. 1.15).

Для зміни позиції камери скористаємось її властивістю `position`.

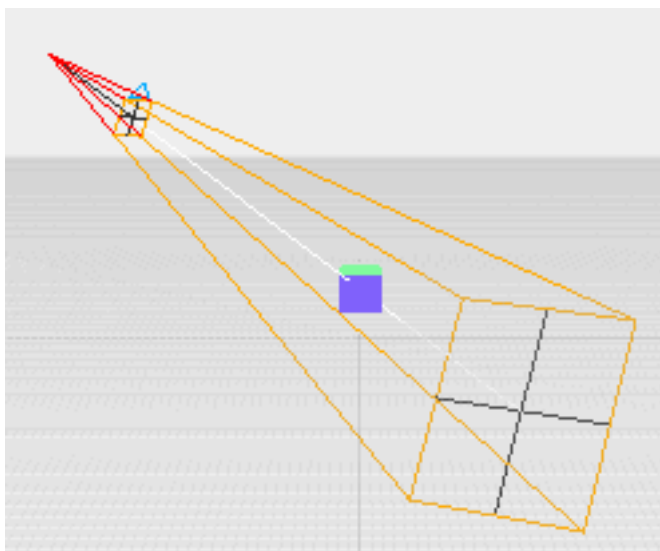


Рис. 1.15. Зрізана піраміда огляду.

Рендерер – це те, що буде відображати 3D-модель на полотні з урахуванням матеріалу, текстури та освітлення. Змінна `renderer` створюється динамічно за допомогою виклику `new` як об'єкт класу `WebGLRenderer`. Однойменна функція-конструктор класу в якості параметру приймає об'єкт у форматі JSON. Для роботи WebAR додатків важливо, щоб сцена була прозорою – тоді на неї можна буде накласти відеопотік з камери. Це досягається встановленням значенням параметру `alpha` у `true`.

Повний перелік властивостей та методів класу `WebGLRenderer` доступні у документації; в якості прикладу використано метод `setSize` встановлення висоти та ширини полотна (`canvas`) – площини, на яку проєціюється сцена.

Безпосередньо рендеринг виконує метод `render`, який відображає проєкцію сцени на холст із точки зору камери.

Останній крок – зв'язування полотна зі сторінкою HTML – виконується викликом

```
document.body.appendChild(renderer.domElement);
```

Отже, ми використовуємо елемент `canvas` для відображення результатів рендерингу (рис. 1.16).

Перед зв'язування полотна зі сторінкою HTML для WebAR додатків необхідно виконати підключення відеопотоку. Для цього до документу HTML додамо тег (об'єкт) `video`:

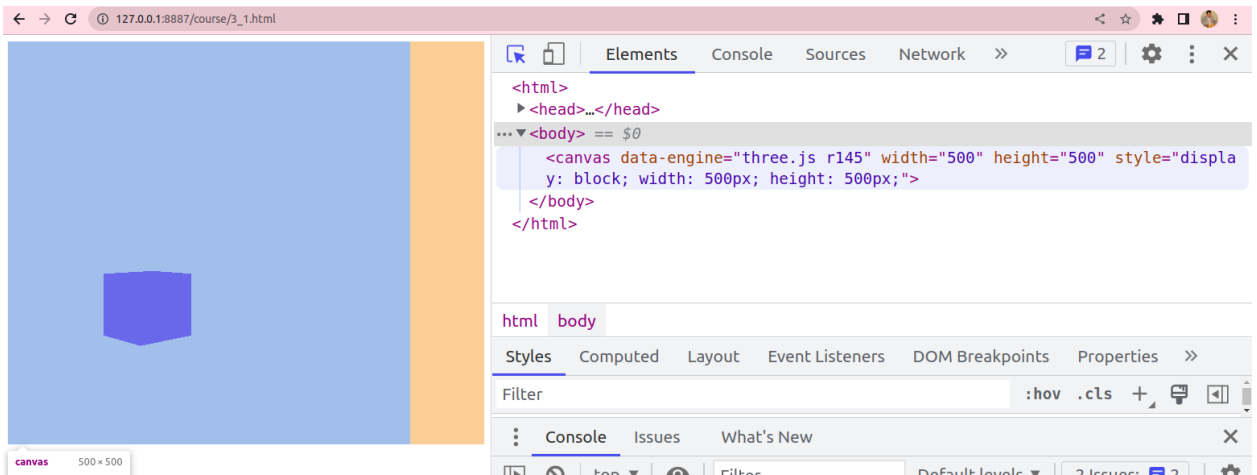


Рис. 1.16. Об'єкт `canvas`, убудований у документ HTML.

```
const video = document.createElement("video");
```

Усередині об'єкту `video` розмістимо відеопотік з камери:

```
navigator.mediaDevices.getUserMedia({video:true})  
  .then((stream) => {  
    video.srcObject = stream;  
    video.play();  
  });
```

`Navigator` є інтерфейсом доступу до стану та властивостей певного веб-браузера. Отримати відповідний об'єкт можна зверненням до властивості вікна `window.navigator`. Метод `mediaDevices` повертає посилання на об'єкт класу `MediaDevices`, що може бути використаний для отримання інформації про доступні медіапристрої (`enumerateDevices`), визначити їх властивості (`getSupportedConstraints`) та отримати доступ до них (`getUserMedia`).

Якщо виклик `getUserMedia` із запитом лише на відео (`video:true`) буде успішним, то буде отримано посилання на відеопотік `stream`, який необхідно пов'язати із властивістю `srcObject` об'єкту `video`. Останнім викликається метод `play` для початку відтворення (програвання) відеопотоку з камери у об'єкті `video`.

Налаштуємо елементи каскадної таблиці стилів (CSS) для об'єктів `video` та `renderer`:

```
video.style.position = "absolute";
video.style.width = renderer.domElement.width;
video.style.height = renderer.domElement.height;
renderer.domElement.style.position = "absolute";
```

Значення параметру `position` встановлюється у `absolute` – такий елемент “щезає” з того місця, де він мав бути й позиціонується заново. Усі інші елементи розташовуються так, нібито цього елемента ніколи не було, а ширина елемента встановлюється за його вмістом. У нашому випадку ширина та висота об’єкту `video` встановлені у аналогічні значення полотна, на якому працює `renderer`, тому застосування `position: absolute` як до `video`, так й до `renderer` надає можливість сумістити (накласти) ці два об’єкти.

Додати створений та налаштований об’єкт `video` до документу необхідно до додавання `renderer.domElement` – тоді зображення куба буде поверх відеопотоку:

```
document.body.appendChild(video);
```

На рис. 1.17 показано першу реалізацію WebAR, в якій реальний об’єкт з камери доповнений віртуальним об’єктом.

Розміщення полотна поверх відео є основою WebAR. Єдине, чого бракує, це відображення об’єкту у більш доцільному місці та оновлення його положення відповідно до сигналу з камери, тобто відстеження об’єкту.

1.2.3. Налаштування бібліотеки для доповненої реальності у Web

Змінити положення зображення можна шляхом переміщення віртуальної камери, змінюючи її позицію (координати) та нахил. Доцільні зміни вимагають відстеження об’єктів, тому поширеною є класифікація доповненої реальності на маркерну, безмаркерну, координатну тощо.

Автор бібліотеки MindAR пропонує класифікацію доповненої реальності за типом відстеження.

Перший тип – *відстеження зображень*: у цьому типі віртуальні об’єкти з’являються поверх цільових зображень, які можуть бути маркер-



Рис. 1.17. Результат накладання.

ними (barcode-like), які мають заздалегідь визначену структуру, та природними, які можуть бути чим завгодно.

Зображення не обов'язково має бути друківаними чи екранними – можуть бути навіть футболки з доповненою реальністю [66].

Другий тип доповненої реальності – *відстеження обличчя*, за якого об'єкти прикріплюються до людського обличчя. Прикладами є фільтри в Instagram, Google Meet, кампанії у соціальних мережах, додатки для примірки віртуальних аксесуарів тощо.

Третій тип доповненої реальності – *відстеження довкілля* (world tracking), який також називають безмаркерною доповненою реальністю. За такого типу відстеження об'єкти доповненої реальності можуть бути розміщені де завгодно, не обмежуючись конкретним зображенням, обличчям або фізичними об'єктами.

Додатки відстеження довкілля безперервно фіксують і відстежують навколишнє середовище та оцінюють фізичне положення користувача до-

датку. Найчастіше об'єкти доповненої реальності прикріплюються до певної поверхні, зокрема, до землі.

Геокоординатна доповнена реальність (location-based AR), відома за Pokémon GO, Ingress тощо передбачає прив'язку контенту до певного географічного положення – широти та довготи. Зазвичай ці програми відстежують доквілля, оскільки доповнений вміст, як правило, прикріплений до землі, а геокоординатна частина є скоріше додатковою умовою, виконання якої приводить до початку відстеження доквілля (або обличчя) у певному місці.

Можуть бути визначені й інші типи відстеження – відстеження 3D-об'єктів, відстеження рук та ін.

Незважаючи на різноманіття бібліотек для доповненої реальності, їх основною задачею є визначення позиції віртуальної камери відповідно до відстежуваного об'єкту, що ілюструється наступним псевдокодом:

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

Спочатку необхідно ініціювати бібліотеку – певний AR-рушій, та отримати посилання на нього. Далі у безперервному циклі дочекатись кадру з відеопотоку реальної камери, визначити її положення (координати на нахил) та перемістити віртуальну камеру на полотні у те саме положення.

Нерідко, однак, рухають не віртуальну камеру, а об'єкти на сцені – тоді визначається положення не реальної камери, а відстежуваного об'єкту, після чого об'єкт віртуальної реальності переміщується у те саме положення, що й відстежуваний об'єкт:

```
const ar = new SOME_AR_ENGINE();
while(true)
```

```

{
  await nextVideoFrameReady();
  const {position, rotation} = ar.computeObjectPose(video);
  cube.position = position;
  cube.rotation = rotation;
}

```

Зображення, що відстежується, може бути будь-якого походження, проте воно має бути підготовленим: так, якщо воно містить зайві елементи (рис. 1.18, праворуч), їх необхідно видалити (рис. 1.18, ліворуч).



Рис. 1.18. Вихідне та підготовлене зображення.

Для розпізнавання зображення з використанням бібліотеки MindAR необхідно виділити на ньому опорні точки – елементи, за якими буде виконуватись розпізнавання. Це можна зробити за допомогою компілятора зображень, розміщеного за посиланням <https://hiukim.github.io/mind-ar-js-doc/tools/compile>.

Результатом роботи компілятора є бінарний файл `targets.mind`, що містить опис опорних точок (рис. 1.19), відстеження яких відбуватиметься.

Інші бібліотеки мають схожі засоби отримання опису зображень, які часто називають компіляторами NFT маркерів (від natural feature tracking – відстеження природних зображень). Таке зображення повинно бути візуально складним та мати високу роздільну здатність (тут деталі мають



Рис. 1.19. Візуалізація опорних точок.

значення). Візуально складне зображення надає програмному забезпеченню багато можливостей для відстеження унікальних частин зображення, що легко розпізнаються.

Від фізичного розміру NFT маркера також залежить якість його розпізнавання – до малих за розміром зображень мобільний пристрій необхідно наблизити, у той час як від великих навпаки – тримати подалі.

Якість розпізнавання також залежить від освітленості екрану мобільного пристрою; крім того, камери з низькою роздільною здатністю зазвичай працюють краще, коли вони знаходяться близько до маркерів.

Після завантаження файлу `targets.mind` перейменуємо його на `altabor.mind` та створимо мінімальний файл HTML:

```
<html>
  <head>
    <meta name="viewport"
      content="width=device-width, initial-scale=1.0">
    <script src=
      "https://cdn.jsdelivr.net/npm/mind-ar/dist/mindar-image-three.prod.js"
    ></script>
```



```

<script src="./main.js"></script>
<style>
  html, body {
    position: relative;
    margin: 0;
    width: 100%;
    height: 100%;
    overflow: hidden
  }
</style>
</head>
<body>
</body>
</html>

```

Згідно рекомендацій у документації MindAR, останню версію бібліотеки MindAR для спільного відстеження зображень і роботи із Three.js (`mindar-image-three`) можна отримати з CDN за універсальним посиланням <https://cdn.jsdelivr.net/npm/mind-ar/dist/mindar-image-three.prod.js> або вказавши номер версії бібліотеки: <https://cdn.jsdelivr.net/npm/mind-ar@1.1.5/dist/mindar-image-three.prod.js>.

Стиль документу налаштований на повновіконне зображення. Так само, як і в попередньому прикладі, файл `main.js` міститиме мінімально необхідний для роботи код JavaScript:

```

const THREE = window.MINDAR.IMAGE.THREE;

document.addEventListener("DOMContentLoaded", () => {

});

```

Бібліотека Three.js є частиною MindAR (модуль `MINDAR.IMAGE.THREE` прикріплений до об'єкта `window`), тому для доступу до неї створюється синонім `THREE`. Наступний код розміщується усередині обробника події `DOMContentLoaded`:


```
const start = async() => {  
  
}  
start();
```

Подія `DOMContentLoaded` опрацьовується тоді, коли завершено завантаження документа HTML. Проте певний час може бути необхідний для того, щоб налаштувати веб-камеру та підготувати її до роботи. Для того, щоб дочекатись результатів певної функції, перед нею ставиться ключове слово `await`, яке не можна використати у звичайних функціях, тому у середині обробника створена та викликається асинхронна функція `start` із наступним вмістом:

```
const mindarThree = new window.MINDAR.IMAGE.MindARThree({  
  container: document.body,  
  imageTargetSrc: "altabor.mind",  
});  
const {renderer, scene, camera} = mindarThree;  
  
const geometry = new THREE.PlaneGeometry(1, 1);  
const material = new THREE.MeshBasicMaterial({  
  color: 0x00ffff, transparent: true, opacity: 0.5  
});  
const plane = new THREE.Mesh(geometry, material);  
  
const anchor = mindarThree.addAnchor(0);  
anchor.group.add(plane);  
  
await mindarThree.start();  
renderer.setAnimationLoop(() => {  
  renderer.render(scene, camera);  
});
```

`mindarThree` є об'єктом класу `MindARThree`, що має два параметри: `container` – місце, асоційоване з відеопотоком (у нашому випадку це тіло

документа HTML), та `imageTargetSrc` – шлях до файлу опису опорних точок відстежуваного зображення (результат роботи компілятора зображень з розширенням `.mind`).

Конструктор класу `MindARThree` має також три додаткові параметри, `uiError`, `uiScanning` та `uiLoading` – встановлення їх значень у `no` надасть можливість відключити початковий екран (подробиці – у документації: <https://hiukim.github.io/mind-ar-js-doc/examples/custom-ui>).

`MindARThree` також створює необхідні для роботи із `Three.js` об'єкти – рендерер, сцену та камеру, доступні відповідно як поля `renderer`, `scene` і `camera` об'єкту `mindarThree`.

Далі створюється об'єкт `plane`, що визначається геометрією `PlaneGeometry` (площина одиничного розміру) та матеріалом із власним кольором `MeshBasicMaterial` (напівпрозорий ціанового кольору).

`anchor` – якірний об'єкт, що повертається викликом методу `addAnchor`, параметр якого відповідає номеру зображення, що розпізнається. Через те, що опрацьовується лише одне зображення, використовується його номер (0 – перше зображення, 1 – друге і т. д.). Якірні об'єкти використовуються для відстеження цільових зображень та надають позицію, в якій повинен бути розміщений об'єкт.

Замість того, щоб додавати площину безпосередньо до сцени, вона додається до складової якоря – об'єкту `group` класу `THREE.Group`, що визначає множину пов'язаних об'єктів, положенням, орієнтацією та видимістю яких можна керувати спільно. Ця якірна група управляється бібліотекою, яка буде постійно оновлювати положення і орієнтацію групи відповідно до нашого набору для відстеження.

Метод `start` об'єкту `mindarThree` і є тим, для якого довелося зробити однойменну зовнішню функцію асинхронною. Це метод не лише виконує налаштування параметрів та вмикання камери (цього можна було й не чекати), а й завантажує у пам'ять веб-браузера усі необхідні дані (у нашому випадку – файл `altabor.mind`).

Для того, щоб рендерер, камера та сцена запрацювали, необхідно створити функцію для їх візуалізації. У безіменній функції зворотного виклику, що створюється функцією `setAnimationLoop`, для кожного кадру із об'єкту `renderer` викликається метод `render`, параметрами якого є об'єкти

scene та camera – це і є анімація на полотні.

У результаті отримуємо повнофункціональний WebAR додаток, що відстежує одне зображення (рис. 1.20).

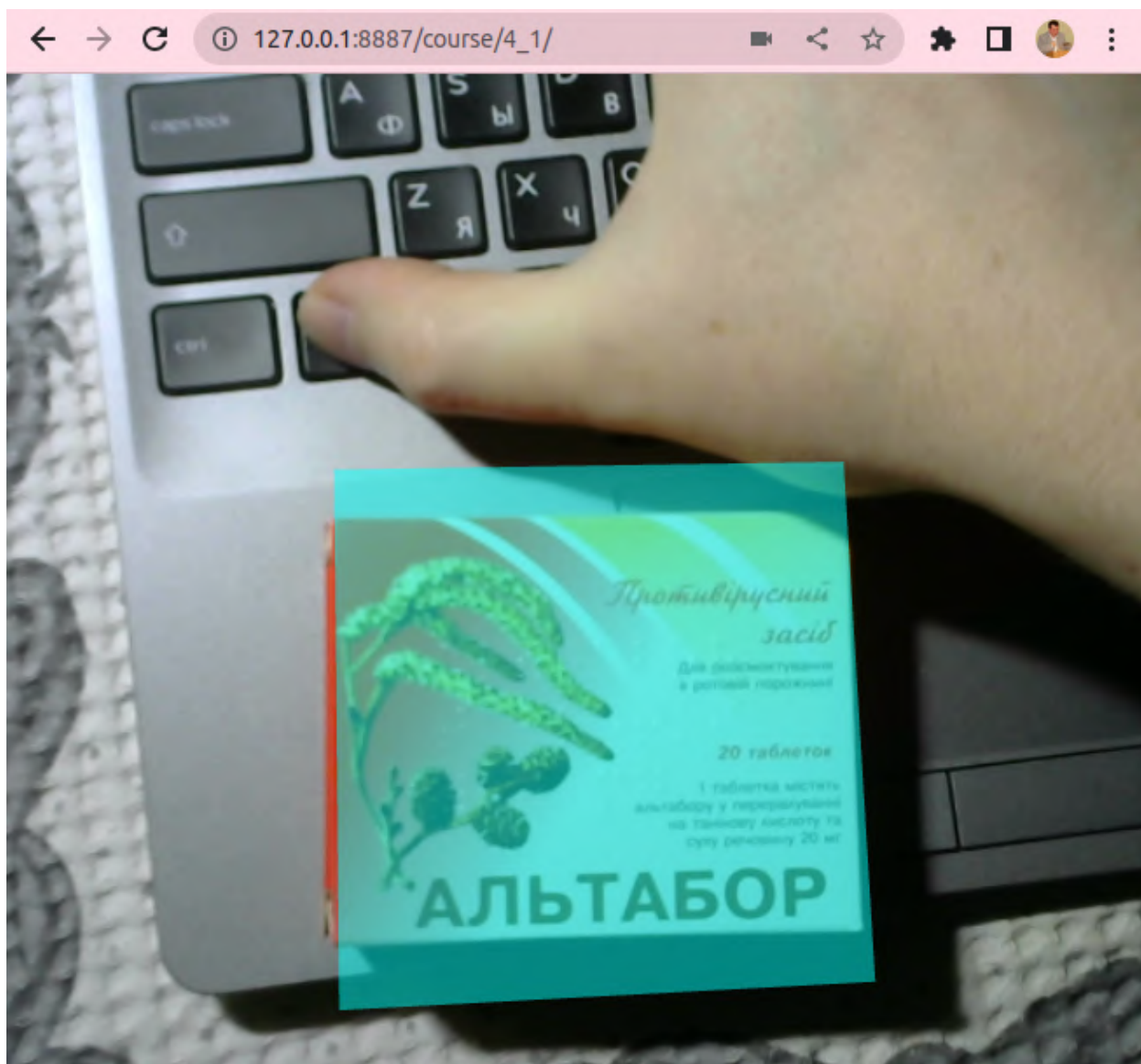


Рис. 1.20. Результат розпізнавання зображення.

Незважаючи на те, що відстежуване зображення є прямокутником, площа масштабувалась до нього за більшою стороною, тому вибір її розмірів як одиночної є цілком виправданим. Для повного перекривання реального зображення віртуальним об'єктом необхідно встановити його розміри пропорційно до розмірів реального: найбільший розмір (у нашому випадку – ширина) в 1, а найменший – у число між 0 та 1 (у нашому випадку – висота: 0.8).

Висновки до розділу 1

1. Для виконання систематизованого бібліометричного аналізу за запитамі “WebAR” та “Web-based augmented reality for education” був застосований VOSviewer версії 1.6.18. В якості джерела даних для першого запиту був обраний Crossref із пошуком по заголовках документів, що надало можливість відібрати 19 документів 2017–2022 рр. Відібрані документи були проаналізовані за кількістю разів, які вони цитуються разом із іншими документами. Із 92 джерел, що цитувались у 19 документах, 26 цитуються спільно більше одного разу, утворюючи лише 1 кластер, що складають роботи, виконані під керівництвом С. О. Семерікова.
2. В якості джерела даних для другого запиту був обраний Scopus із пошуком по заголовках, анотаціях та ключових словах документів, що надало можливість відібрати 93 документи 2001–2023 рр., 66 з яких припадають на останні 5 років. Більшу частину з них складають статті у журналах (58), меншу – книги (4) та статті у матеріалах конференцій (31). Аналіз розподілу ключових слів надав можливість виділити 4 кластери:
 - (а) перший кластер пов’язує базові концепти доповненої реальності в освіті: доповнену та віртуальну реальність із освітою (зокрема, медичною) та навчанням людей, зокрема, із використанням смартфонів;
 - (б) другий кластер пов’язує концепти проєктування навчального середовища: викладання, інженерну освіту, комп’ютерно зорієнтоване викладання, електронне навчання, студентів, мобільне навчання, навчальні середовища, ІКТ навчання та курсукулуми;
 - (в) третій кластер пов’язує концепти реалізації імерсивного навчального середовища: веб-сайти, мотивацію, системи навчання, інтерактивні навчальні середовища, AR-додатки та технології доповненої реальності;
 - (г) четвертий кластер містить концепт дистанційної освіти,

пов'язаний у першому кластері із концептами доповненої та віртуальної реальності і концептом освіти (включно з медичною), у другому – із концептами студента, інженерної освіти, ІКТ навчання, електронного навчання та курикулуму, та у третьому – із концептом веб-сайту.

3. Аналіз розподілу концептів за густиною зв'язків та часом надає можливість визначити, що найбільш давні (до 2015 року) дослідження були зосереджені на питаннях інтерфейсів користувача та їх застосування в освіті. У роботах 2016 року центр уваги зміщувався у напрямі дослідження впливу на студентів викладання у навчальних середовищах. Дослідження 2017 року актуалізували концепти віртуальної реальності, інтерактивних навчальних середовищ, курикулуму, комп'ютерно зорієнтованого навчання, зокрема, у інженерній освіті. Центром уваги досліджень 2018 року стали ІКТ навчання, застосування смартфонів, AR-додатків та педагогічних технологій доповненої реальності. WebAR стає предметом уваги досліджень 2019 року – у них підіймаються питання застосування смартфонів, інтернет/веб-орієнтованого навчання та доповненої реальності. У 2020 року вплив пандемії COVID-19 додав питання мотивації навчання та медичної освіти. Новий елемент останніх за часом досліджень – це доповнення (аугментація) людини.

4. Дібрані засоби розробки доповненої реальності для Web утворюють три групи:

(а) основні засоби:

- веб-сервер Web Server for Chrome надає необхідну повну функціональність без потреби встановлення, що задовольняє вимогам простоти та мобільності;
- засіб перенаправлення трафіку ngrok надає можливість доступу до веб-сервера, який знаходиться за брандмауером (на учнівському або вчительському комп'ютері), що надає можливість спільної роботи у віддаленому режимі;
- віддалений налагоджувач RemoteJS надає можливість ви-

конувати налагодження програм мовою JavaScript на мобільних пристроях, послуговуючись десктопними браузерами;

- (б) графічна бібліотека Three.js є високорівневою реалізацією крос-платформного стандарту відображення WebGL мовою JavaScript, що надає можливість працювати із графічними абстракціями високого рівня;
- (в) бібліотека для доповненої реальності MindAR надає можливість роботи із природними зображенням в якості якорів доповненої реальності та містить у своєму складі бібліотеки Three.js і TensorFlow.js.

Останнє є ключовим для інтеграції моделей машинного навчання, створених за допомогою TensorFlow, із WebAR додатками, побудованими із використанням MindAR.

РОЗДІЛ 2

ЕЛЕМЕНТИ МЕТОДИКИ НАВЧАННЯ РОЗРОБКИ ДОПОВНЕНОЇ РЕАЛЬНОСТІ ДЛЯ WEB ІЗ ІНТЕГРОВАНИМИ МОДЕЛЯМИ МАШИННОГО НАВЧАННЯ

2.1. Methodика розробки WebAR додатків для відстеження обличчя

2.1.1. Модель опорних точок обличчя

Бібліотека MindAR має два основні набори модулів – для роботи з зображеннями (`image`) та для роботи з обличчями (`face`), – що можуть використовуватись одночасно, але потребують окремого підключення у документі HTML:

```
<html>
  <head>
    <meta name="viewport"
      content="width=device-width, initial-scale=1.0">
    <script src=
      "https://cdn.jsdelivr.net/npm/mind-ar/dist/mindar-face-three.prod.js"
    ></script>
    <script type="importmap">
      {
        "imports": {
          "three": "https://unpkg.com/three/build/three.module.js"
        }
      }
    <script src="./main.js"></script>
    <style>
      html, body {position: relative; margin: 0;
        width: 100%; height: 100%; overflow: hidden}
    </style>
```

```
</head>
<body>
</body>
</html>
```

Схожість API для відстеження зображень і відстеження обличчя чітко простежується у коді `main.js`. Так, спочатку у аналогічний спосіб відбувається створення посилання на об'єкти Three.js, але не з `window.MINDAR.IMAGE.THREE`, а з `window.MINDAR.FACE.THREE`:

```
const THREE = window.MINDAR.FACE.THREE;
```

Так само всі дії виконуються у функції `start`, що виключається після завантаження документу HTML. Відмінність полягає у застосуванні конструктора класу `MindARThree` з `window.MINDAR.FACE.THREE` замість `window.MINDAR.IMAGE`. Через те, що бібліотека розпізнає обличчя, параметр `imageTargetSrc` для цього конструктора не потрібен:

```
document.addEventListener('DOMContentLoaded', () => {
  const start = async() => {
    const mindarThree = new window.MINDAR.FACE.MindARThree({
      container: document.body,
    });
  };
});
```

Для створення об'єктів Three.js застосовуються властивості об'єкта `mindarThree`: рендерер, сцена, камера:

```
const {renderer, scene, camera} = mindarThree;
```

Створимо якір та прив'яжемо до нього напівпрозору сферу:

```
const geometry = new THREE.SphereGeometry(0.1, 32, 16);
const material = new THREE.MeshBasicMaterial({color: 0x00ffff,
  transparent: true, opacity: 0.5});
const sphere = new THREE.Mesh(geometry, material);

const anchor = mindarThree.addAnchor(1);
```



```
anchor.group.add(sphere);

await mindarThree.start();
renderer.setAnimationLoop(() => {
  renderer.render(scene, camera);
});
}
start();
});
```

Попри схожість, метод `addAnchor` по-іншому трактує параметр: якщо для `window.MINDAR.IMAGE` це був номер цільового зображення, то при розпізнаванні обличчя це буде номер опорної точки обличчя. Точка 1 – це ніс (рис. 2.1).



Рис. 2.1. Прив'язування об'єкту до опорної точки обличчя.

Виявлення опорних точок обличчя базується на відомій моделі бібліотеки TensorFlow [44]. Модель MediaPipe Face Mesh [48] є згортковою нейронною мережею, що визначає на обличчі 468 тривимірних опор-

них точок (https://github.com/tensorflow/tfjs-models/raw/master/face-landmarks-detection/mesh_map.jpg), і ми можемо прив'язувати об'єкти до будь-якої з них (рис. 2.2).

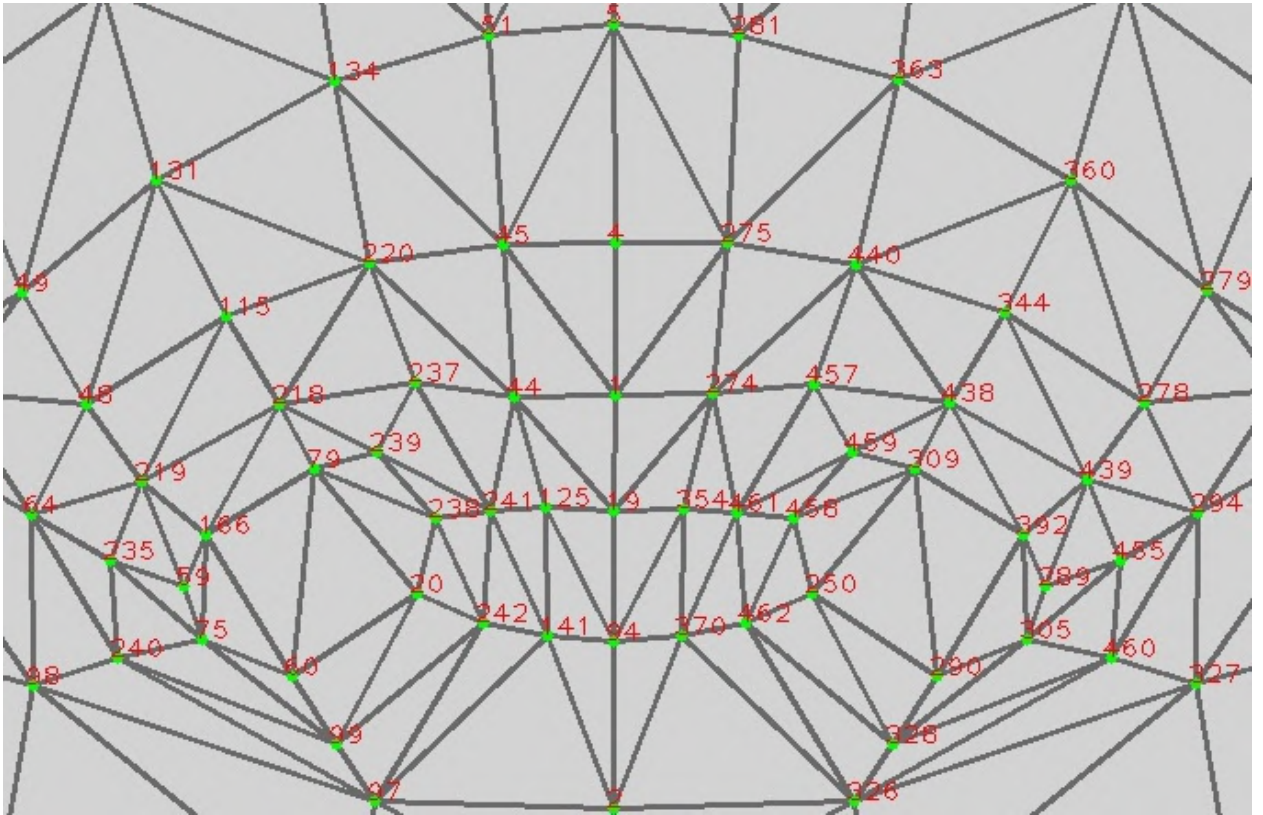


Рис. 2.2. Опорні точки обличчя (фрагмент).

2.1.2. Накладання маски на обличчя

Маска для обличчя (face mesh) – ще один тип доповненої реальності, пов'язаної з накладання зображень (текстур) на всі опорні точки обличчя людини, а не прив'язки до окремих із них. Маски для обличчя використовуються для створення різноманітних ефектів макіяжу, татуювання тощо – аж до повної віртуалізації особи.

Маска для обличчя не є заздалегідь визначеною 3D-моделлю – вона динамічно генерується з постійним оновленням геометрії.

Для накладання маски на обличчя нам знадобиться відповідна текстура, для завантаження якої застосовується метод `loadTexture` з користувачької бібліотеки `loader.js` (додаток А):

```
import {loadTexture} from "./loader.js";
```

Створення маски відбувається викликом `addFaceMesh`:

```
const faceMesh = mindarThree.addFaceMesh();
```

Метод `addFaceMesh` за формою схожий на `addAnchor`, але сутність у них різна: у `addAnchor` створюється порожня група, до якої додаються об'єкти, положенням яких керує MindAR, у той час як `faceMesh`, що повертається `addFaceMesh`, є єдиним відображуваним об'єктом, геометрія якого змінюється у кожному кадрі.

Матеріалом маски для обличчя може бути довільна текстура – якщо її не встановити, маска для обличчя буде виглядати, як показано на першому зображенні (рис. 2.3).

Побачити структуру цієї маски можна на другому зображенні (рис. 2.3) – для цього необхідно встановити атрибут `wireframe` матеріалу зображення:

```
faceMesh.material.wireframe = true;
```

Третє, четверте та п'яте зображення (рис. 2.3) є прикладами накладання модифікованих та оригінальної (https://github.com/google/mediapipe/blob/master/mediapipe/modules/face_geometry/data/canonical_face_model_uv_visualization.png) текстур опорних точок обличчя. У документації до Meta Spark Studio [115] можна знайти набір текстур для масок для обличчя, що можна використовувати для створення власних масок, як описано у [107].

В якості прикладу застосуємо текстуру з [90] (рис. 2.3, шосте зображення):

```
const texture = await loadTexture("moustache1.png");
faceMesh.material.map = texture;
```

Текстура `texture`, встановлена в якості матеріалу маски для обличчя, містить певні прозорі елементи, для підтримки яких встановлюється властивість матеріалу `transparent` (рівень прозорості регулюється значенням `opacity`):

```
faceMesh.material.transparent = true;
```

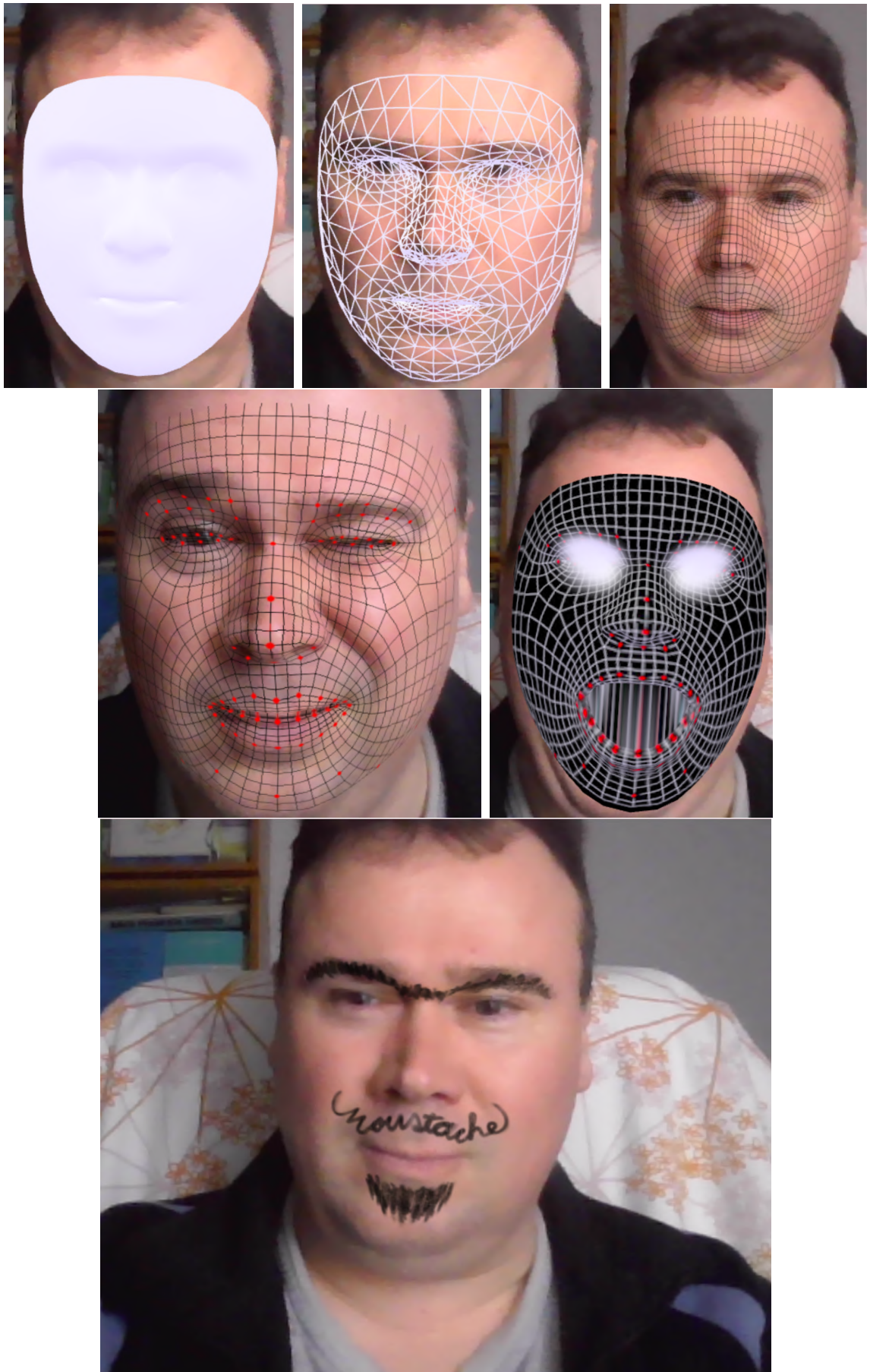


Рис. 2.3. Маски для обличчя.

Встановлення параметру `needsUpdate` необхідно для вказання на необхідність оновлення текстури при зміні геометрії:

```
faceMesh.material.needsUpdate=true;
```

Останній крок – розміщення маски на сцені; в ідеалі, маска для обличчя повинна мати таку ж геометрію та положення, як і саме обличчя:

```
scene.add(faceMesh);
```

Створення гарної маски потребує певних художніх навичок, проте, послуговуючись канонічною текстурою (рис. 2.2), це зробити досить просто – достатньо нанести поверх неї необхідне зображення та видалити зайві лінії.

2.2. Методика інтеграції моделей машинного навчання у WebAR додатки

2.2.1. Інтеграція стандартних моделей

Для машинного навчання у Інтернеті найчастіше використовують TensorFlow [102] – безкоштовну бібліотеку машинного навчання з відкритим вихідним кодом, розроблену компанією Google. На сьогодні вона підтримує багато мов, включаючи основні – Python, Java, C++ – та підтримувані спільнотою: Haskell, C#, Julia, R, Ruby, Rust, Scala. Вона доступна на багатьох платформах, включаючи Linux, Windows, Android, а також убудованих платформах – версія бібліотеки TensorFlow Lite призначена для роботи з моделями машинного навчання на мобільних пристроях, мікроконтролерах, пристроях Інтернету речей тощо.

TensorFlow.js [104] – версія TensorFlow на JavaScript, що надає можливість розробляти та використовувати моделі, послуговуючись цією мовою, безпосередньо у браузері.

TensorFlow.js поставляється з великою кількістю попередньо навчених моделей, які можна одразу використовувати [105]. Повний перелік моделей, доступних на поточний момент, подано за посиланням <https://>

[//github.com/tensorflow/tfjs-models](https://github.com/tensorflow/tfjs-models) – багато із них є надзвичайно корисними і можуть стати гарним доповненням до AR-додатків. Якщо необхідна функціональність відсутня, можна створити та навчити власні моделі, або модифікувати наявні.

Розглянемо найпростіший приклад інтеграції моделі TensorFlow у AR-додаток з використанням бібліотеки MindAR. Для цього створимо наступний документ:

```
<html>
  <head>
    <meta name="viewport"
      content="width=device-width, initial-scale=1.0">
    <script src=
"https://cdn.jsdelivr.net/npm/mind-ar/dist/mindar-image-three.prod.js"
    ></script>
    <script>
      window.tf = window.MINDAR.IMAGE.tf;
    </script>
    <script src=
"https://unpkg.com/@tensorflow-models/handpose/dist/handpose.js">
    </script>
    <script src="./main.js" type="module"></script>
    <style>
      html, body {position: relative; margin: 0; width: 100%;
        height: 100%; overflow: hidden}
    </style>
  </head>
  <body>
  </body>
</html>
```

Першим підключається бібліотека MindAR – точніше, та її частина, що відповідає за виявлення та відстеження зображень: `mindar-image-three.prod.js`. `TensorFlow.js` є її частиною, тому єдина необхідна дія – це створення властивості `tf` вікна `window` як копії

`window.MINDAR.IMAGE.tf`. Якби бібліотека `MindAR` не використовувалась, то підключення `TensorFlow.js` виконувалося б у такий спосіб:

```
<script src=
"https://cdn.jsdelivr.net/npm/@tensorflow/tfjs/dist/tf.min.js"
></script>
<script src=
"https://unpkg.com/@tensorflow/tfjs-core/dist/tf-core.js"></script>
<script src=
"https://unpkg.com/@tensorflow/tfjs-converter/dist/tf-converter.js"
></script>
```

Моделі не є частиною `Tensorflow.js`, тому їх необхідно підключати окремо – так, як показано на прикладі моделі `handpose.js`, описаної у [51]. Ця модель використовується для визначення кисті руки та її складових.

Файл `main.js` містить основний код програми, на початку якої визначається асинхронна функція `start`, що виконується після того, як завершується завантаження документа HTML:

```
const THREE = window.MINDAR.IMAGE.THREE;

document.addEventListener("DOMContentLoaded", () => {
  const start = async() => {
    // вміст функції start
  }
  start();
});
```

Першим створюється примірник AR-рушія (об'єкт `mindarThree`), що ідентифікуватиме зображення, задане набором опорних точок `altabor.mind`, у відеопотоці в тілі документа:

```
const mindarThree = new window.MINDAR.IMAGE.MindARThree({
  container: document.body, imageTargetSrc: "altabor.mind",
});
```

Об'єкт `mindarThree` містить необхідні для подальшої роботи властивості:

```
const {renderer, scene, camera} = mindarThree;
```

`plane` є квадратною напівпрозорою площиною ціанового кольору розміром 1×1 (у одиницях ширини зображення):

```
const geometry = new THREE.PlaneGeometry(1, 1);
const material = new THREE.MeshBasicMaterial({
  color: 0x00ffff, transparent: true, opacity: 0.5
});
const plane = new THREE.Mesh(geometry, material);
```

`anchor` – якір, пов'язаний із відстежуваним зображенням:

```
const anchor = mindarThree.addAnchor(0);
```

Додавання площини до групи об'єктів, пов'язаної з якорем, забезпечує її відображення, коли бібліотека знаходить відстежуване зображення у відеопотоці:

```
anchor.group.add(plane);
```

Наступний крок – завантаження моделі `handpose` з TensorFlow Hub [103]: переглянувши цей репозитарій моделей, можна побачити, що вони займають чималий обсяг, тому метод `load`, що їх завантажує, викликається як асинхронна функція:

```
const model = await handpose.load();
```

Наступний крок – запуск AR-рушія:

```
await mindarThree.start();
```

Цикл анімації розділимо на дві частини: перша буде включати лише базовий рендеринг та налаштовується викликом `setAnimationLoop`:


```
renderer.setAnimationLoop(() => {
  renderer.render(scene, camera);
});
```

Модель `handpose` опрацьовує окремі кадри, які беруться з відеопотоку `video`. Це достатньо обчислювально ємна процедура, тому, урахуваючи, що, доки велика точність ідентифікації рук непотрібна, можна спробувати виявляти їх не у кожному кадрі, а, наприклад, у кожному десятому. Для цього використовується лічильник кадрів `frameCount`:

```
const video = mindarThree.video;

let frameCount = 1;
```

Функція `detect` утворює другий, більш змістовний, цикл анімації:

```
const detect = async () => {
```

Для кожного десятого кадру виконується виклик методу `estimateHands` завантаженої моделі, якому передається кадр `video`:

```
  if (frameCount % 10 == 0) {
    const predictions = await model.estimateHands(video);
```

Метод повертає масив `predictions`, який містить відомості про детектовані у кадрі зображення рук, тому ненульовий розмір масиву – ознака того, що у кадрі була рука:

```
    if (predictions.length > 0) {
```

На жаль, дійсно поки що лише одна рука – поточна версія моделі дає можливість визначити лише одну руку, тому звернемось до першого елемента масиву `predictions` для отримання даних про неї. Серед багатьох детектованих властивостей, що містить цей елемент, є, наприклад, `boundingBox`, елементами якого є `topLeft` та `bottomRight` – відповідно лівий верхній та правий нижній кути прямокутника, що обмежує руку. Відповідно, `w` та `h` є відносними розмірами руки – вони будуть дорівнювати 1 лише у випадку, коли рука займатиме весь кадр:

```

const x = predictions[0].boundingBox.topLeft[0];
const y = predictions[0].boundingBox.topLeft[1];
const x1 = predictions[0].boundingBox.bottomRight[0];
const y1 = predictions[0].boundingBox.bottomRight[1];
const w = (x1 - x) / window.innerWidth,
      h = (y1 - y) / window.innerHeight;

```

Якщо рука є видимою, зробимо видимим й площину `plane`, масштабувавши її за відносними розмірами руки:

```

plane.visible = true;
plane.scale.set(w, h, 1);

```

Положення площини на детектованому зображенні встановимо так, щоб воно відображало положення обмежувального прямокутника руки у кадрі (рис. 2.4):

```

plane.position.set(x/window.innerWidth,
                  y/window.innerHeight, 0);
}

```

Якщо руки у кадрі немає, площину не показуватимемо:

```

else
  plane.visible = false;
}

```

Наприкінці функції `detect` викликом `requestAnimationFrame` повідомляємо браузеру, що йому буде необхідно оновити полотно саме за допомогою цієї функції, та збільшуємо на 1 лічильник кадрів:

```

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

```

Остання дія у функції `start` – запуск другого циклу анімації викликом `requestAnimationFrame`:

```

window.requestAnimationFrame(detect);

```

Створений ефект достатньо простий, проте він надає уявлення про те, як використовувати моделі машинного навчання у AR-додатках.

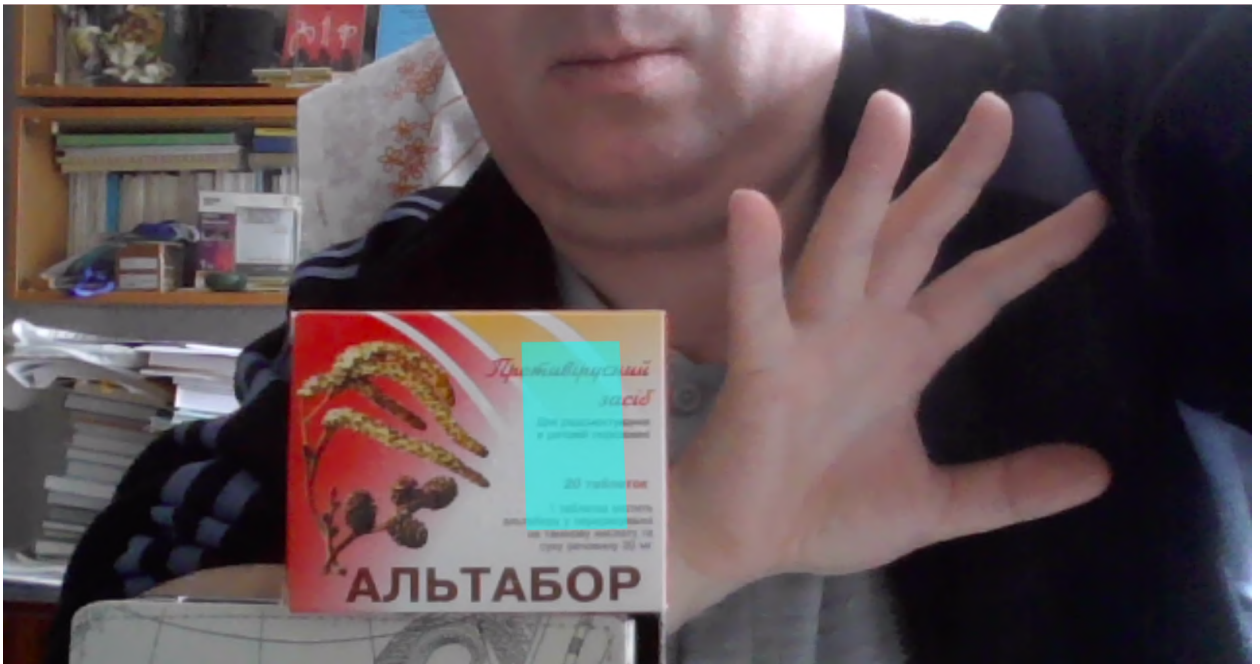


Рис. 2.4. Жестове управління розміром та положенням віртуального об'єкта.

2.2.2. Розробка користувацьких моделей

Для швидкого створення та навчання власної моделі можна скористатись Teachable Machine [100] – складовою проєкту Google A.I. Experiment, що надає можливість будувати моделі для розв'язання задач класифікації зображень, звуків та поз. В якості прикладу розглянемо адаптований план уроку з використанням Teachable Machine, розроблений Ян Чень [22; 88]:

Тема уроку: Teachable Machine

Рівень: 6-8 клас

Основне питання: Як штучний інтелект (ШІ) класифікує дані?

Огляд уроку: Учні вивчать Teachable Machine і дізнаються про концепції машинного навчання, класифікації та їх суспільний вплив.

Міжпредметні зв'язки:

- з мовою та літературою:

- писати інформаційні/пояснювальні тексти для вивчення теми та чітко передавати ідеї та інформацію;
 - ставити запитання та відповідати на запитання щодо інформації, отриманої від співрозмовника, пропонуючи відповідні деталі;
 - визначати значення слів і словосполучень, які використовуються в текстах шкільного рівня.
- з природничими науками:
 - розпізнавати та використовувати процес технологічного проектування для вирішення проблем.

Словник уроку:

- *Машинне навчання*: здатність комп'ютерів навчатися без явного програмування.
- *Контрольоване навчання (навчання з учителем)*: тип машинного навчання, де алгоритми навчаються на основі маркованих даних.
- *Дерева рішень*: тип контрольованого машинного навчання, де дані розбиваються за певним параметром.

План уроку:

1. Розминка (5 хвилин)
2. Розминка з Teachable Machine (10 хвилин)
3. Гра “Класифікація риб” – навчання учнів (15 хвилин)
4. Гра “Класифікація риб” – навчання машин (10 хвилин)
5. Підведення підсумків і завершення (10 хвилин)

Дидактичні цілі:

- *Основні*
 1. Я можу описати вхідні та вихідні дані класифікаційної моделі.

2. Я можу пояснити, як ШІ класифікує дані шляхом розпізнавання образів.

- *Додаткові*

3. Я можу виразити результати машинного навчання у термінах достовірності.

4. Я можу створювати дерево рішень за допомогою операторів IF, THEN.

5. Я можу бачити, як штучний інтелект впливає на суспільство.

Обладнання та матеріали:

- веб-браузер із доступом до веб-камери;
- роздатковий матеріал “Класифікація риб” (<https://edu.readyai.org/wp-content/uploads/2020/03/Fish-Game-Handout.pdf>);
- зображення риб (поділитись із учнями або через Диск Google (<http://bit.do/AIfishgame>), або скопіювавши файли на комп’ютери, до яких надається спільний доступ);
- робочий аркуш “Дерево рішень” (<https://edu.readyai.org/wp-content/uploads/2020/03/Decision-Tree-Worksheet.pdf>);
- олівці.

Хід уроку

Розминка (5 хвилин)

Мета: познайомити учнів із поняттям класифікації.

- Короткий огляд, “що таке ШІ”.
 - Машини, які демонструють людський інтелект, або, розумні машини.
- Що таке класифікація?

- Подумайте про час, коли вам потрібно було розкласти речі по категоріях. Наприклад, розділяти білизну за кольорами або сортувати пошту.
- Подібно до вас, ШІ може навчитися передбачати, до якої категорії належать дані.
 - * ШІ може створити розбивку групи людей на невеликі групи чоловіків і жінок;
 - * ШІ може розкладати листи в папки “Спам” і “Не спам”.
- Як навчається штучний інтелект?
 - Сьогодні ми дізнаємося про підхід, який називається *контрольованим навчанням*, коли алгоритми навчаються на основі маркованих даних. Це означає, що ми надаємо алгоритму дані, вже позначені правильними категоріями.

Поради учителю. В рамках машинного навчання існує два основних типи задач: контрольовані і неконтрольовані. Основна відмінність полягає в тому, що в контрольованому навчанні ми заздалегідь знаємо, якими повинні бути вихідні значення для наших зразків. З іншого боку, неконтрольоване навчання не має маркованих результатів, тому його мета полягає в тому, щоб зробити висновок про природну структуру, присутню в даних.

Розминка з Teachable Machine (10 хвилин)

Мета: познайомити учнів із Teachable Machine.

- Об’єднайте учнів у пари. Перейдіть на сайт <https://teachablemachine.withgoogle.com/>.
 - Натисніть кнопку *Get Started*.
 - Оберіть *Image Project*.
 - Оберіть *Standard Image Model*.
 - Додаткову допомогу щодо роботи із Teachable Machine можна знайти тут: <https://www.youtube.com/playlist?list=PLJfHZtseuscuTQfodmFnbZ3rBgCWSRT9t>.

- Навчальна діяльність.

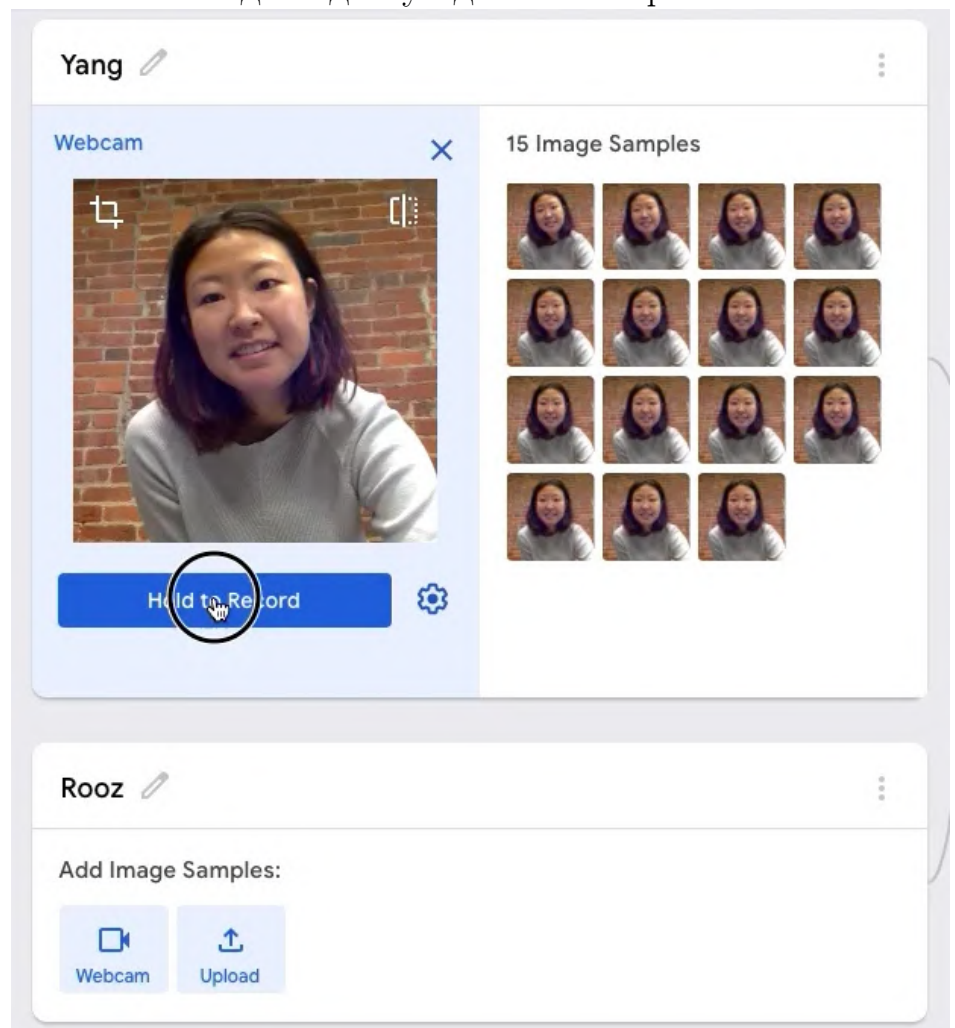
- Чи може Teachable Machine знати, що ви і ваш партнер – різні люди?

- * Замість *Class 1* уведіть ім'я першої особи.

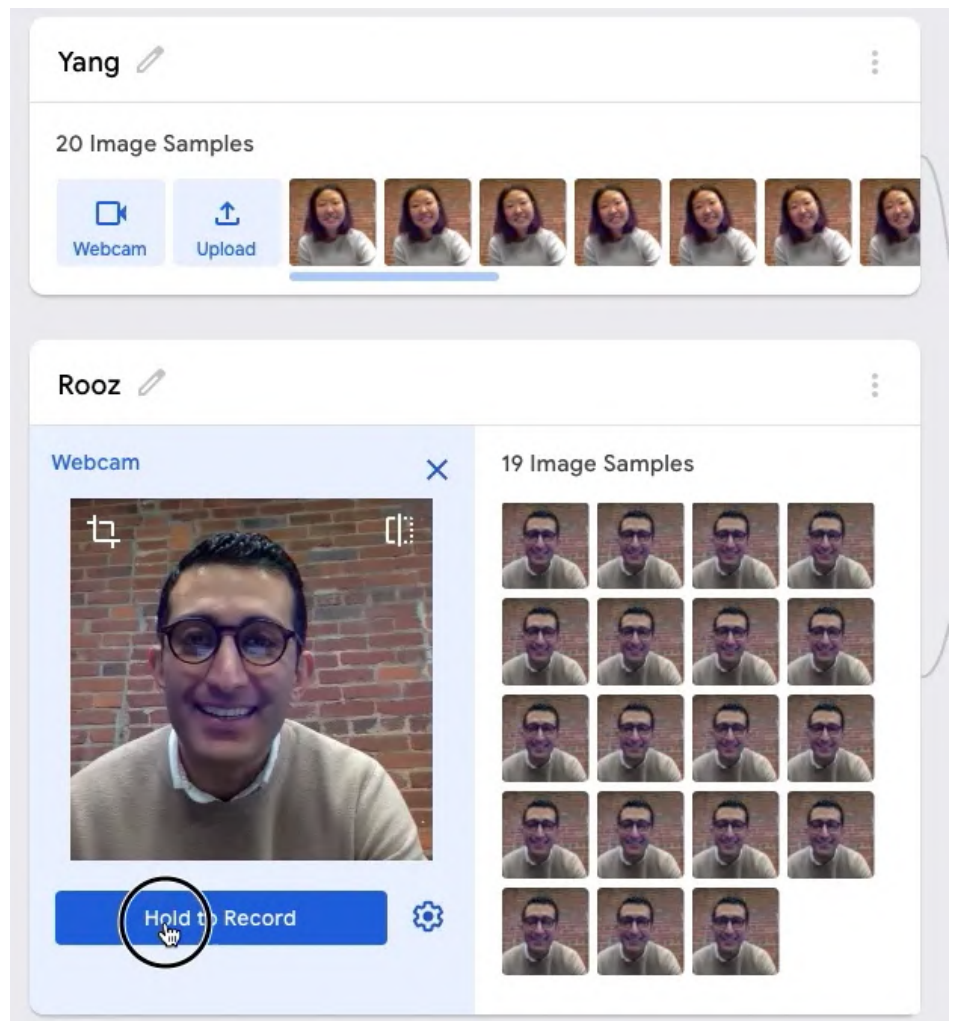
- * Замість *Class 2* уведіть ім'я другої особи.

- * Демонстрація

- Натисніть *Webcam* і зробіть 20 зображень себе під першим класом (перевизначеним як ваше ім'я). *Примітка:* ви повинні надати доступ до веб-камери.



- Зробіть 20 зображень другої людини під другим класом (перевизначеним як її ім'я).



- Дозвольте учням завершити цей крок, перш ніж перейти до наступного кроку.

Поради учителю. Кількість зображень може вплинути на результат. Ми повернемося до цього питання іншим разом, але зараз важливо визначити і дотримуватися кількості зображень, які необхідно зробити.

- Навчання моделі.
 - Натисніть *Train*.
 - Примітка: Після того, як модель закінчила навчання, настав час її протестувати.
 - * Як ви думаєте, чи навчилася модель?
- Оцінка результатів.

- Чи може Teachable Machine успішно розпізнати вас і вашого партнера?
- Спробуйте змінити фон, вираз обличчя або позу. Чи все ще працює модель?
- *Додаткова справа:* Змініть розмір вхідних даних і подивіться, чи впливає це на точність? Що ще може вплинути на точність? Чому?
- *Додаткова справа:* Якщо модель стає менш точною, коли ви стоїте на іншому фоні, спробуйте повернутися назад і додати більше ваших фотографій на новому фоні та перенавчити модель.

Поради учителю.

1. Якщо учні знайдуть випадки, коли модель працює не так, як очікувалося, це чудова можливість зрозуміти, як працює машинне навчання.
2. Коли ви даєте комп'ютеру зображення, все, що “бачить” модель – це величезні сітки пікселів. Оскільки модель не має розуміння людини, те, що має значення для моделі, може бути не тим, що ви думаєте. Наприклад, алгоритм може вловлювати відмінності у фонах або освітленні, а не у ваших обличчях.

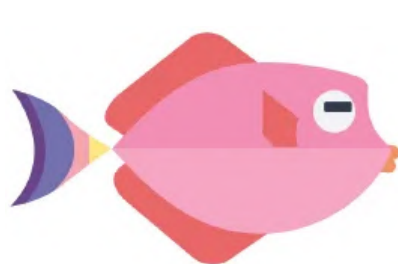
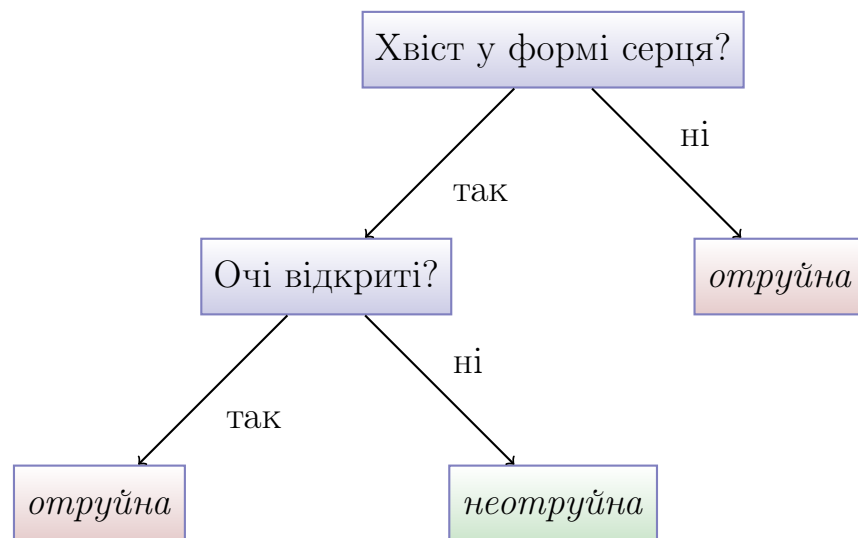
Гра “Класифікація риб” – навчання учнів (15 хвилин)

Мета: учні дізнаються про класифікацію, придумуючи правила прийняття рішень щодо отруйних/неотруйних риб.

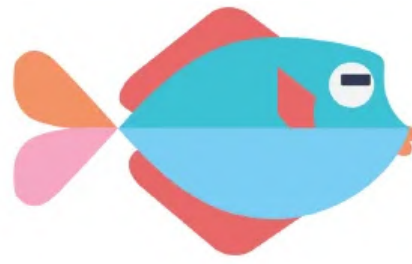
- Закрийте ноутбуки для безмашинної частини уроку.
- Ми надали Teachable Machine два набори даних і сказали їй, які з них належать до якої категорії. Це було контрольоване чи неконтрольоване навчання? Чому?
 - Контрольоване навчання. Ми позначили дані і сказали машині, який набір даних належить до якого класу. Таким чином, машина могла визначити категорію нових даних.

- Роздайте кожній групі учнів роздатковий матеріал “Класифікація риб” (<https://edu.readyai.org/wp-content/uploads/2020/03/Fish-Game-Handout.pdf>).
- Ми – вчені-екологи, перед якими стоїть завдання класифікувати рибу як отруйну або неотруйну. Деякі фізичні характеристики риби визначають, до якої категорії вона потрапляє. Яка ваша гіпотеза? (Дидактична ціль 1)
 - Запропонуйте студентам провести невеликий мозковий штурм. Вони можуть порівняти малюнки в парах.
 - Запропонуйте учням придумати правила прийняття рішень, використовуючи робочий аркуш “Дерево рішень” (<https://edu.readyai.org/wp-content/uploads/2020/03/Decision-Tree-Worksheet.pdf>).

Дерево рішень [КЛЮЧ УЧИТЕЛЯ]



Отруйна риба



Неотруйна риба

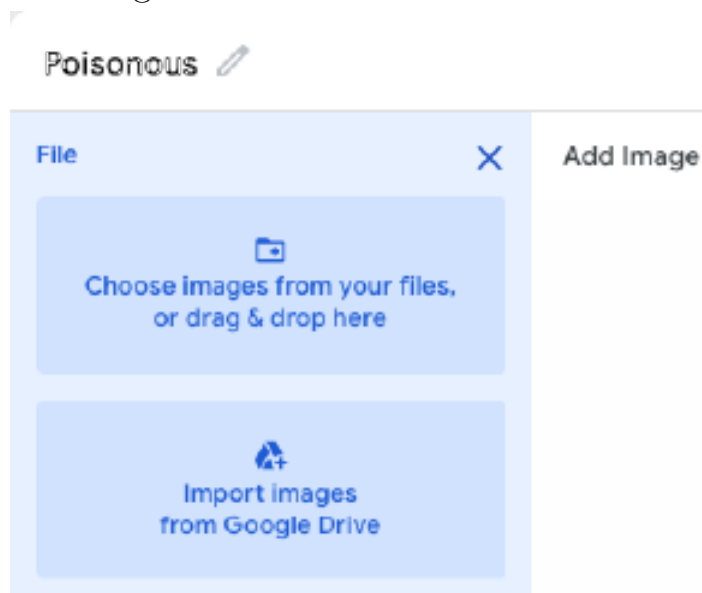
- Роздайте учням олівці та тестові дані (остання сторінка роздаткового матеріалу).

- Учні повинні роздивитись тестові дані і зробити прогнози щодо того, чи є ці нові риби отруйними або неотруйними, виходячи зі своєї гіпотези.

Гра “Класифікація риб” – навчання машин (10 хвилин)

Мета: учні перевіряють свої теорії за допомогою Teachable Machine.

- Поверніться до комп’ютерів, оновіть сторінку, щоб розпочати новий проєкт з розпізнавання зображень.
- Навчальна діяльність.
 - Чи може Teachable Machine визначити, яка риба отруйна, а яка неотруйна?
 - * Перейменуйте *Class 1* у Poisonous (отруйні).
 - * Перейменуйте *Class 2* у Non-Poisonous (неотруйні).
 - * Завантажте зображення для кожного класу за допомогою кнопки *Upload*. Оберіть завантаження з локального диска або з Диска Google:



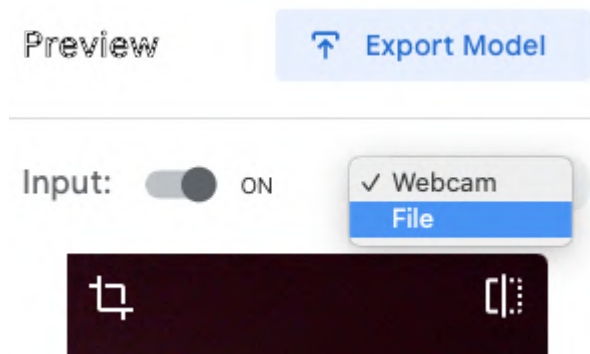
- * Навчіть вашу модель.
 - Натисніть *Train*.

Поради учителю. Найпростіший спосіб надати спільний доступ до цих зображень – це поділитися посиланням на Диск Google (<http://bit.ly>).

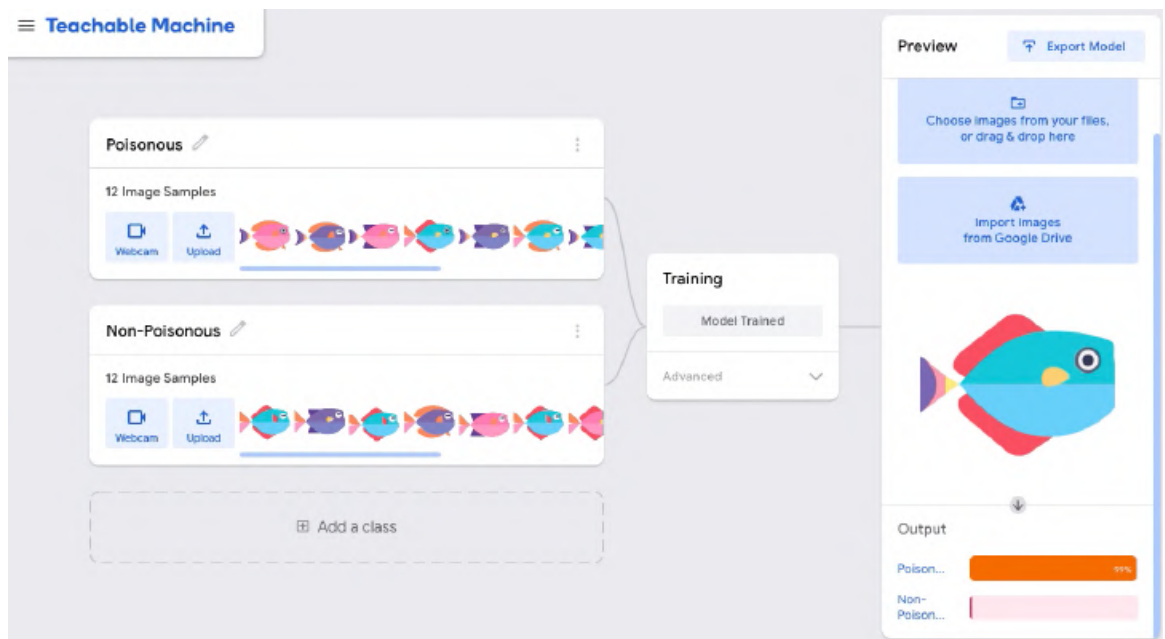
do/AIfishgame), а потім попросити учнів скористатися опцією “Import Images Using Google Drive”.

- Тестування.

- Завантажте тестові дані (4 зображення риб) одне за одним. Порівняйте результати Teachable Machine з вашими власними висновками з попередньої вправи. Чи погоджується модель з вами?



Скористайтесь опцією “File” для завантаження

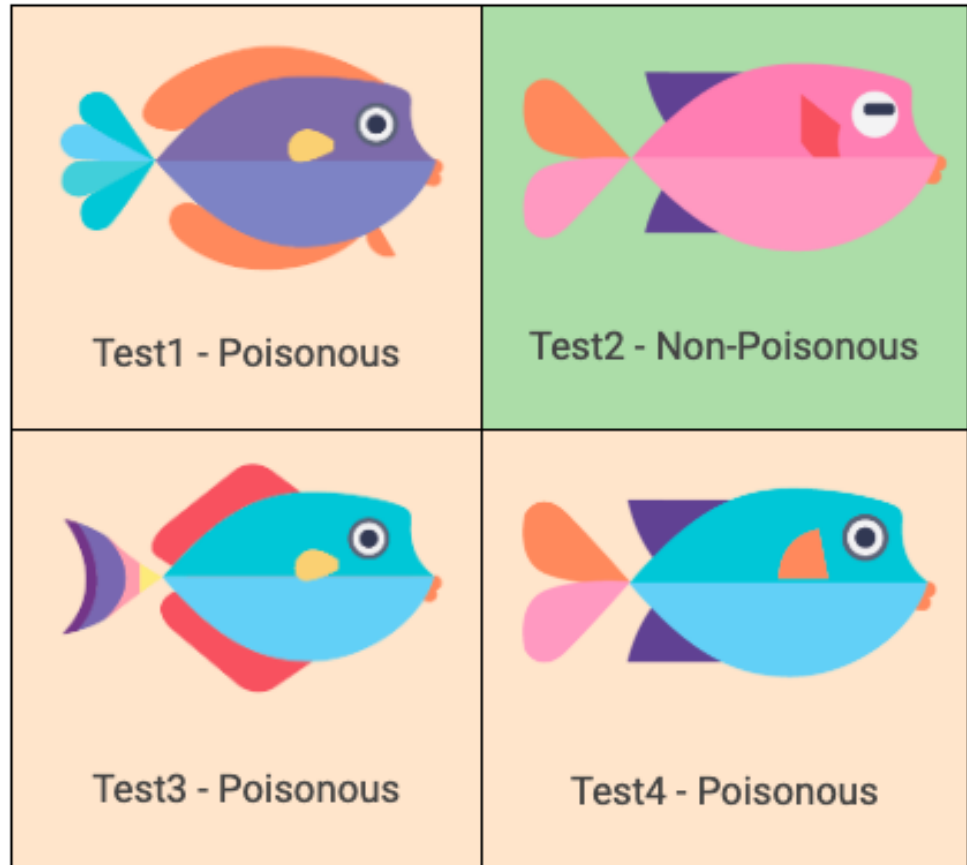


Тестуємо нову рибу з моделлю

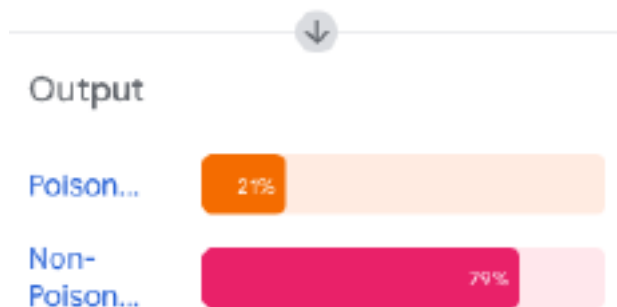
- Оцінка результатів.

- Наскільки успішно Teachable Machine розпізнає отруйні та неотруйні риби?

[КЛЮЧ УЧИТЕЛЯ]



- Іноді модель неправильно розпізнає Рибу 4. Скористайтеся цією можливістю, щоб проаналізувати навчальну вибірку і поміркуйте, чому Риба 4 є особливо складною для моделі.



Алгоритм неправильно вважає, що Риба 4 неотруйна

- Можливо, синій колір тіла Риби 4 менш представлений у множині отруйних риб, тоді як Риба 2 добре представлена серед неотруйних, а це означає, що є багато інших риб, схожих на Рибу 2.
 - Крім того, навчальний набір дуже малий (усього 12 зображень!), що ускладнює розпізнавання образів і робить здогадки моделі менш точними.
- *Додаткова вправа:* Машинне навчання використовує довірчі інтервали, що варіюються від 0% до 100%. Це дуже важлива концепція, оскільки повсякденні програми машинного навчання мають тенденцію приховувати цей факт, вказуючи відповіді як певні або дискретні. Teachable Machine стверджує, що вона на 78% впевнена, що визначила рибу як отруйну, але в повсякденному застосуванні це означає результат “так” або “ні”. (*Дидактична ціль 3*)

Підведення підсумків і завершення (10 хвилин)

Мета: Учні повторюють та оцінюють.

- Сьогодні ми дізналися про машинне навчання, спочатку спробувавши класифікувати самі, а потім навчивши машинну модель робити те ж саме.
- Люди і машини навчаються, розпізнаючи образи. Подібно до того, як ви всі вчитеся, коли вчитель виправляє ваші математичні помилки, машини також піддаються навчанню. Наприклад, контрольоване машинне навчання схоже на навчання на прикладах: комп'ютеру надається набір даних, що містить мітки, які виступають в ролі відповідей. З часом машина може, по суті, “навчитися” розрізняти ці мітки для отримання правильного результату.
- Як ви думаєте, які ще програми використовують ШІ у такий спосіб у повсякденному житті?
 - Голосові помічники, такі як Alexa або Siri.
 - Сортування спаму і неспаму в електронній пошті.

- *Додаткова справа:* Які додатки ви бачите в майбутньому, що використовують контрольоване навчання і класифікацію? (*Дидактична ціль 5*).
 - Класифікація оточення для безпілотних автомобілів.
 - Ідентифікація пухлин на рентгенівських знімках.
 - Сортування вторинної сировини від сміття.
 - Виявлення шахрайських операцій з кредитними картками.
 - Ідентифікація рослин за фотографіями.

Локалізовані версії роздаткового матеріалу “Класифікація риб” та робочий аркуш “Дерево рішень” можна знайти у додатках Бта Ввідповідно. Застережемо, однак, від застосування кирилиці у іменах категорії – це може призвести до подальших проблем із експортом моделі.

2.2.3. Інтеграція користувацьких моделей

Бібліотеки, що входять до складу Teachable Machine, базуються на моделях TensorFlow: MobileNet для класифікації зображень [74], Speech Commands для класифікації звуків [98] та PoseNet для класифікації поз тіла [81].

Відповідно, побудована модель класифікації риб може бути експортована та застосована так само, як і раніше використані моделі опорних точок обличчя та пози рук.

Натискання кнопки “*Export Model*” надає можливість експорту у форматах:

- TensorFlow.js – розміщення моделі за посиланнями [https://teachablemachine.withgoogle.com/models/\[...\]](https://teachablemachine.withgoogle.com/models/[...]) або завантаження моделі та коду JavaScript і p5.js (рис. 2.5);
- TensorFlow – завантаження коду Python та моделі у форматах h5 (Keras) та Savedmodel (TensorFlow);
- TensorFlow Lite – завантаження моделі у форматі tflite для пристроїв IoT на базі Android та Coral.

Export your model to use it in projects. ✕

Tensorflow.js ⓘ Tensorflow ⓘ Tensorflow Lite ⓘ

Export your model:

Upload (shareable link) Download [Download my model](#)

Code snippets to use your model:

Javascript p5.js [Contribute on Github](#)

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() {

```

Copy

Рис. 2.5. Завантаження моделі для TensorFlow.js.

Архів з моделлю для TensorFlow.js містить 3 файли:

- `metadata.json` – текстовий файл у форматі JSON, що містить відомості про номери версій TensorFlow.js (`tfjsVersion`), Teachable Machine (`tmVersion`), бібліотеки зі складу Teachable Machine (`packageVersion`) та її імені (`packageName` – у нашому випадку це `@teachablemachine/image`), дату створення (`timeStamp`) та ім'я моделі (`modelName` – за замовчанням `tm-my-image-model`), розмір зображення (`imageSize` – усі зображення приводяться до одного розміру) та категорії (`labels`), що використовувались для розмітки даних: `Poisonous`, `Non-Poisonous`;
- `model.json` – текстовий файл у форматі JSON, що містить відомості про архітектуру нейронної мережі (`modelTopology`);
- `weights.bin` – бінарний файл, що містить вагові коефіцієнти нейрон-

ної мережі.

Розглянемо незвичну, але потужну техніку, що може бути використана у WebAR додатках – використання елементів HTML та CSS для AR-контенту, таких як `div`, `image`, `p`, що може бути прикріплений до цільових якорів.

```
<html>
  <head>
    <meta name="viewport" content="
      width=device-width, initial-scale=1.0">
    <script type="importmap">
      {
        "imports": {
          "three": "https://unpkg.com/three/build/three.module.js"
        }
      }
    </script>
    <script src=
"https://cdn.jsdelivr.net/npm/mind-ar/dist/mindar-image-three.prod.js"
  ></script>
    <script>
      window.tf = window.MINDAR.IMAGE.tf;
    </script>
    <script src="https://cdn.jsdelivr.net/npm/@teachablemachine/
image/dist/teachablemachine-image.min.js"></script>
    <script src="./main.js" type="module"></script>

  <style>
    html, body { position: relative; margin: 0; width: 100%;
      height: 100%; overflow: hidden }
    #ar-div { width: 1000px; height: 1000px; visibility: hidden;
      background: #0000ff; opacity: 0.6; font-size: 100px;
      display: flex; justify-content: center;
      align-items: center; color: white; }
```

```
    </style>
</head>
<body>
  <div id="ar-div"></div>
</body>
</html>
```

При експорті моделей пропонується тестовий код для їх перевірки, з якого можна дізнатись, як підключити бібліотеку `teachablemachine-image.min.js` через CDN.

`div` є частиною документа HTML – контейнером, що описується стилем CSS та може містити що завгодно. У MindAR для об'єктів Three.js значення 1 відповідає найбільшому розміру цільового зображення, проте для об'єктів CSS це значення – 1000, тому для контейнеру `ar-div` встановлені такі стильові налаштування: ширина (`width`) та висота (`height`) – 1000 пікселів (1000px), при першому зверненні контейнер є невидимим (`visibility: hidden`), колір фону (`background`) – синій (`#0000ff`), колір елементів контейнеру (`color`) – білий (`white`), прозорість (`opacity`) – 60% (0.6), розмір шрифту надписів (`font-size`) – 100 пікселів, розташування елементів у контейнері (`display`) динамічне (`flex`), елементи контейнеру розташовуються (`justify-content`), починаючи з центру (`center`), та вирівнюються (`align-items`) за центром.

Three.js та MindAR мають можливість рендерингу HTML та елементів CSS, проте у MindAR вона є вбудованою, у той час як у Three.js необхідно підключити додатковий модуль `CSS3DRenderer`.

`CSS3DRenderer` можна використовувати для ієрархічних 3D-перетворень DOM-елементів через властивість CSS3 `transform`. Цей рендерер особливо цікавий, якщо ви хочете застосувати 3D-ефекти на сайті без рендеринга на основі полотна. Він також може бути використаний для того, щоб об'єднати DOM-елементи з WebGL-контентом.

Однак із `CSS3DRenderer` неможливо використовувати систему матеріалів та геометрії Three.js, тому що `CSS3DRenderer` орієнтований на звичайні DOM-елементи. Ці елементи загортаються в спеціальні об'єкти (`CSS3DObject` або `CSS3DSprite`) і потім додаються до сцени.

`CSS3DObject` не є частиною Three.js API, тому доводиться

імпортувати його явно. Це, у свою чергу, вимагає імпортування бібліотеки Three.js для того (яка вже є частиною MindAR). Для того, щоб не імпортувати одну й ту ж бібліотеку двічі (причому цілком ймовірно, що у різних версіях), у документ HTML додано таблицю імпорту імен. Визначення імені `three` виконується для задоволення залежностей класу `CSS3DObject` у наступному коді:

```
import {CSS3DObject} from "https://unpkg.com/three/examples/"+
"jasm/renderers/CSS3DRenderer.js";

const THREE = window.MINDAR.IMAGE.THREE;

document.addEventListener("DOMContentLoaded", () => {
  const start = async() => {
    const mindarThree = new window.MINDAR.IMAGE.MindARThree({
      container: document.body, imageTargetSrc: "altabor.mind",
    });
    const {renderer, cssRenderer, scene, cssScene, camera}
      = mindarThree;

    const ardiv = document.querySelector("#ar-div");
    const obj = new CSS3DObject(ardiv);
    const cssAnchor = mindarThree.addCSSAnchor(0);
    cssAnchor.group.add(obj);

    const model = await
      tmImage.load("./model.json", "./metadata.json");
    const maxPredictions = model.getTotalClasses();

    await mindarThree.start();
    renderer.setAnimationLoop(() => {
      cssRenderer.render(cssScene, camera);
    });

    let skipCount = 1;
```

```

const video = mindarThree.video;

const detect = async () => {
  if (skipCount % 10 == 0) {
    const prediction = await model.predict(video);

    let str="";
    for (let i = 0; i < maxPredictions; i++)
      str += prediction[i].className + ": "
        + prediction[i].probability.toFixed(2) + "<br>";
    ardiv.innerHTML = str;
  }
  window.requestAnimationFrame(detect);
  skipCount++;
}
window.requestAnimationFrame(detect);
}
start();
});

```

Із об'єкту `mindarThree` можна отримати як звичні об'єкти `renderer` та `scene` класів `WebGLRenderer` та `Scene`, так й `cssRenderer` – об'єкт класу `CSS3DRenderer`, та `cssScene` – ще один об'єкт класу `Scene`.

`obj` є об'єктом класу `CSS3DObject`, що викликом `document.querySelector` пов'язується із визначеним у документі HTML контейнером `ar-div`. Створення цього об'єкту замінює раніше використовувану тріаду “геометрія + матеріал = об'єктна сітка”.

CSS об'єкт `obj` зв'язується зі спеціальним типом якоря – якорем CSS, що створюється викликом `addCSSAnchor`.

У циклі анімації метод `render` об'єкту `cssRenderer` використовується для того, щоб відобразити CSS-сцену (`cssScene`) з точку зору камери (`camera`)

Для звернення до методу завантаження моделі `load` бібліотеки `Teachable Machine` для класифікації зображень використовується ім'я `tmImage`. Параметрами методу є шлях до файлів архітектури та метаданих

моделі – `model.json` та `metadata.json`. Очікується, що файл із ваговими коефіцієнтами буде розміщений там, де знаходиться файл архітектури моделі.

Після завантаження моделі викликом методу `getTotalClasses` можна визначити кількість категорій, що розрізнятиме модель – у нашому випадку це значення, збережене у `maxPredictions`, дорівнюватиме двом: отруйна та неотруйна риби (рис. 2.6).

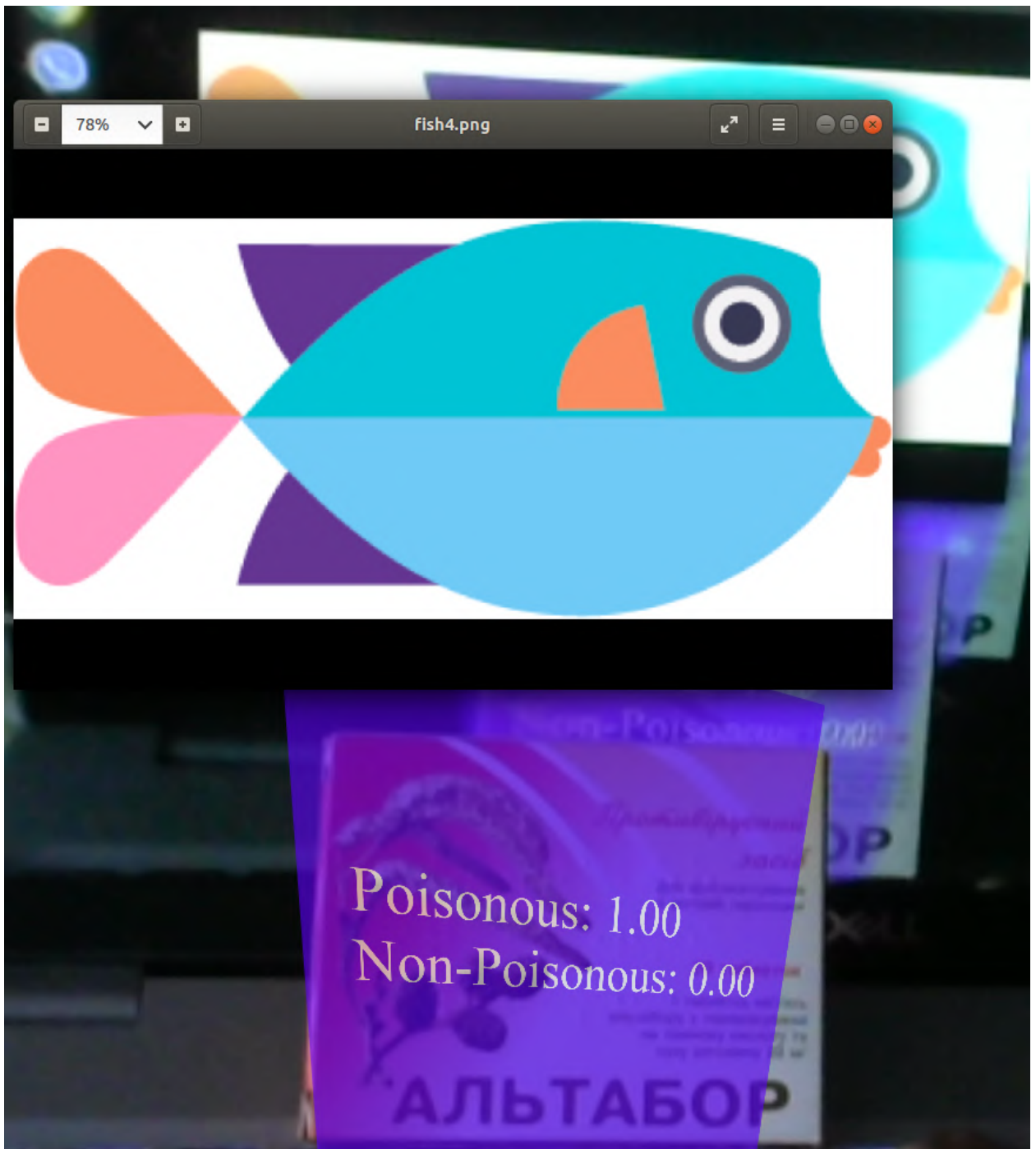


Рис. 2.6. Реалізація гри “Класифікація риб”.

Так само, як і раніше, кожний десятий кадр передаємо моделі для аналізу викликом `predict`, що повертає масив з двох об'єктів, що містять відомості про категорію (`className`) та ймовірність того, що зображення відноситься до неї (`probability`) – рядок із відомостями про них й прикріплюються до елемента CSS `div`.

Із рис. 2.6 можна побачити, що найбільш проблемне зображення (Риба 4) правильно ідентифіковане як отруйна риба з максимальною ймовірністю – це досягається як налаштуванням параметрів навчання моделі (500 ітерацій із точністю 0.0001), так й правильним позиціонуванням зображення перед камерою.

Висновки до розділу 2

1. Розробка WebAR додатків для відстеження обличчя з використанням бібліотеки MindAR передбачає застосування відомої моделі бібліотеки TensorFlow – MediaPipe Face Mesh, яка є згортковою нейронною мережею, що визначає на обличчі 468 тривимірних опорних точок. Бібліотека MindAR надає можливість визначити будь-яку з них в якості якоря, до якого можуть бути прив'язані необхідні об'єкти Three.js та моделі.
2. Повне застосування моделі MediaPipe Face Mesh у бібліотеці MindAR реалізовано у вигляді спеціального якоря – маски для обличчя (`face mesh`), який можна вважати окремим типом доповненої реальності, пов'язаної з накладання зображень (текстур) на всі опорні точки обличчя людини, а не прив'язки до окремих із них. Маски для обличчя використовуються для створення різноманітних ефектів макіяжу, татуювання тощо – аж до повної віртуалізації особи. Маска для обличчя не є заздалегідь визначеною 3D-моделлю – вона динамічно генерується з постійним оновленням геометрії.
3. Методика інтеграції моделей машинного навчання у WebAR додатки передбачає опанування трьох основних кроків:
 - (а) інтеграцію стандартних моделей;
 - (б) розробку користувацьких моделей;

(в) інтеграцію користувацьких моделей.

4. Перший крок передбачає ознайомлення із TensorFlow.js – версією TensorFlow на JavaScript, що надає можливість розробляти та використовувати моделі, послуговуючись цією мовою, безпосередньо у браузері. TensorFlow.js поставляється з великою кількістю попередньо навчених моделей, які можна одразу використовувати – багато із них є надзвичайно корисними і можуть стати гарним доповненням до AR-додатків. На першому кроці доцільно розглянути одну з таких моделей – `handpose.js`, що використовується для визначення кисті руки та її складових, продемонструвати основну проблему WebAR – суттєве падіння продуктивності програми при застосуванні моделі до кожного кадру, та визначити напрями її розв’язання. У результаті першого кроку створюється WebAR додаток для жестового управління розміром та положенням віртуального об’єкта.
5. Якщо необхідна функціональність відсутня у TensorFlow.js, можна створити та навчити власні моделі, або модифікувати наявні. Для цього на другому кроці пропонується застосувати Teachable Machine – складову проєкту Google A.I. Experiment, що надає можливість будувати моделі для розв’язання задач класифікації зображень, звуків та поз. План-конспект уроку інформатики для учнів 6–8 класів, розроблений фахівцями ReadyAI та локалізований українською, надає можливість на прикладі створення гри “Класифікація риб” сформулювати навички, необхідні для побудови власних моделей.
6. Бібліотеки, що складають Teachable Machine, базуються на моделях TensorFlow: MobileNet для класифікації зображень, Speech Commands для класифікації звуків та PoseNet для класифікації поз тіла. Відповідно, на третьому кроці побудована модель класифікації риб може бути експортована та застосована так само, як і раніше використані моделі опорних точок обличчя та пози рук. На цьому кроці розглядається фізична (2 файли у форматі JSON і 1 бінарний файл) та логічна (метадані, архітектура нейронної мережі та її вагові коефіцієнти) структура моделі TensorFlow, та, на основі тестового коду для її перевірки, отриманого у процесі експорту, мо-

дифікується WebAR додаток, розроблений на першому кроці: завантажуються власна модель, визначається кількість категорій, що вона класифікуватиме, а об'єктом доповненої реальності стають відомості про кожну із категорій та ймовірності того, що зображення з веб-камери відноситься до них. Останнє надає можливість для обговорення питань помилок класифікації та їх залежності як від налаштування параметрів навчання моделі, так й способу подання тестових зображень WebAR додатку.

ВИСНОВКИ

Виконане розв'язання проблеми розробки методики навчання розробки доповненої реальності для Web із інтегрованими моделями машинного навчання надало можливість отримати такі висновки:

1. Бібліометричний аналіз джерел за базами Crossref (19 документів 2017–2022 рр.) та Scopus (93 документи 2001–2023 рр.) надав можливість виокремити основні концепти дослідження, згруповані у 4 кластери:
 - (а) перший кластер пов'язує *базові концепти доповненої реальності в освіті*: доповнену та віртуальну реальність із освітою (зокрема, медичною) та навчанням людей, зокрема, із використанням смартфонів;
 - (б) другий кластер пов'язує *концепти проектування навчального середовища*: викладання, інженерну освіту, комп'ютерно зорієнтоване викладання, електронне навчання, студентів, мобільне навчання, навчальні середовища, ІКТ навчання та курикулуми;
 - (в) третій кластер пов'язує *концепти реалізації імерсивного навчального середовища*: веб-сайти, мотивацію, системи навчання, інтерактивні навчальні середовища, AR-додатки та технології доповненої реальності;
 - (г) четвертий кластер містить *концепт дистанційної освіти*, пов'язаний у першому кластері із концептами доповненої та віртуальної реальності і концептом освіти (включно з медичною), у другому – із концептами студента, інженерної освіти, ІКТ навчання, електронного навчання та курикулуму, та у третьому – із концептом веб-сайту.

Аналіз розподілу концептів за густиною зв'язків та часом надав можливість датувати появу різних концептів та відслідкувати їх розвиток від освітніх застосувань інтерфейсів користувача до його аугментації.

2. Дібрані засоби розробки доповненої реальності для Web утворюють три групи:
- (а) основні засоби:
- веб-сервер Web Server for Chrome надає необхідну повну функціональність без потреби встановлення, що задовольняє вимогам простоти та мобільності;
 - засіб перенаправлення трафіку ngrok надає можливість доступу до веб-сервера, який знаходиться за брандмауером (на учнівському або вчительському комп'ютері), що створює умови для спільної роботи у віддаленому режимі;
 - віддалений налагоджувач RemoteJS надає можливість виконувати налагодження програм мовою JavaScript на мобільних пристроях, послуговуючись десктопними браузерами;
- (б) графічна бібліотека Three.js є високорівневою реалізацією кросплатформного стандарту відображення WebGL мовою JavaScript, що надає можливість працювати із графічними абстракціями високого рівня;
- (в) бібліотека для доповненої реальності MindAR надає можливість роботи із природними зображенням в якості якорів доповненої реальності та містить у своєму складі бібліотеки Three.js і TensorFlow.js – останнє є ключовим для інтеграції моделей машинного навчання, створених за допомогою TensorFlow, із WebAR додатками, побудованими із використанням MindAR.
3. У процесі розробки та апробації методики розробки WebAR додатків для відстеження обличчя обґрунтовано доцільність спільного застосування моделі MediaPipe Face Mesh – згорткової нейронної мережі, що визначає на обличчі 468 тривимірних опорних точок, та бібліотеки MindAR, що надає можливість визначити будь-яку з них в якості якоря. Показано, що повне застосування моделі MediaPipe Face Mesh у бібліотеці MindAR реалізовано у вигляді маски для обличчя, що динамічно генерується з постійним оновленням геометрії – типу доповненої реальності, пов'язаного з накладання зображень на всі опорні

точки обличчя людини. Подано приклади застосування масок для обличчя з метою створення ефектів макіяжу, татуювання тощо.

4. Методика інтеграції моделей машинного навчання у WebAR додатки передбачає опанування трьох основних кроків:

- (а) перший крок – *інтеграція стандартних моделей* – передбачає ознайомлення із попередньо навченими моделями TensorFlow.js, що можуть бути застосовані у WebAR додатках. Показано доцільність розгляду моделі handpose.js, що використовується для визначення кисті руки та її складових, продемонстровано основну проблему WebAR – суттєве падіння продуктивності при застосуванні моделі до кожного кадру, та запропоновано спосіб її розв’язання. У результаті першого кроку створюється WebAR додаток для жестового управління розміром та положенням віртуального об’єкта;
- (б) другий крок – *розробка користувацьких моделей* – передбачає створення та навчання власних моделей TensorFlow з використанням Teachable Machine, яка надає можливість будувати моделі для розв’язання задач класифікації зображень, звуків та поз. Наведено план-конспект уроку інформатики для учнів 6–8 класів, у ході якого формуються навички, необхідні для побудови власних моделей;
- (в) третій крок – *інтеграція користувацьких моделей* – виконується шляхом експорту побудованої за допомогою Teachable Machine моделі класифікації риб та модифікації WebAR додатку, розробленого на першому кроці: завантажується власна модель, визначається кількість категорій, що вона класифікуватиме, а об’єктом доповненої реальності стають відомості про кожну із категорій та ймовірності того, що зображення з веб-камери відноситься до них. Останнє надає можливість для обговорення питань помилок класифікації та їх залежності як від налаштування параметрів навчання моделі, так і способу подання тестових зображень WebAR додатку.

Проведене дослідження не вичерпує усіх складових поставленої проблеми – так, потребують подальшого дослідження:

- історія та перспективи розвитку WebAR у освіті;
- методика спільного використання різних середовищ моделювання нейронних мереж;
- розвиток бібліотек WebAR, зокрема – у напрямі реалізації повсюдної доповненої реальності;
- співвідношення реального та віртуального у навчанні в умовах пандемії, стихійного лиха та військового конфлікту.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. 1st International Workshop on Haptic Human-Computer Interaction, HCI 2000 // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) / за ред. S. Brewster, R. Murray-Smith. — 2001. — Т. 2058. — С. 1—218.
2. 2030 Toy web of the future / B. Kang [та ін.] // Advances in Intelligent Systems and Computing. — 2014. — Т. 272. — С. 69—75. — DOI: 10.1007/978-3-319-05570-1_8.
3. 2nd International Conference on Innovative Technologies and Learning, ICITL 2019 // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) / за ред. L. Ronningsbakk [та ін.]. — 2019. — Т. 11937 LNCS.
4. 3DJPi: An open-source web-based 3D simulator for pololu's 3Pi platform / L. O. Maggi [та ін.] //. — Institute of Electrical, Electronics Engineers Inc., 2019. — С. 52—58. — DOI: 10.1109/SVR.2019.00025.
5. A discussion on web-based learning contents with the ar technology and its authoring tools to improve students' skills in exercise courses / T. Kobayashi [та ін.] // / за ред. A. F. Mohd Ayub [та ін.]. — Asia-Pacific Society for Computers in Education, 2014. — С. 34—36.
6. A web-based augmented reality platform using pictorial QR code for educational purposes and beyond / M. Nguyen [та ін.] // / за ред. S. N. Spencer. — Association for Computing Machinery, 2019. — DOI: 10.1145/3359996.3364793.
7. A Web-Based Campus Navigation System with Mobile Augmented Reality Intervention / N. Nordin [та ін.] //. Т. 1997 / за ред. V. Vijejan [та ін.]. — IOP Publishing Ltd, 2021. — DOI: 10.1088/1742-6596/1997/1/012038.
8. *An J., Poly L.-P., Holme T.* Usability Testing and the Development of an Augmented Reality Application for Laboratory Learning // Journal

- of Chemical Education. — 2020. — Т. 97, № 1. — С. 97–105. — DOI: 10.1021/acs.jchemed.9b00453.
9. An Augmented Reality Framework for Gamified Learning / B. Hensen [та ін.] // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) / за ред. R. Klamma [та ін.]. — 2018. — Т. 11007 LNCS. — С. 67–76. — DOI: 10.1007/978-3-319-96565-9_7.
 10. An augmented reality mobile learning experience based on treasure hunt serious game / M. Farella [та ін.] // / за ред. C. Busch [та ін.]. — Academic Conferences, Publishing International Limited, 2021. — С. 148–154. — DOI: 10.34190/EEL.21.109.
 11. *Anwar S., Leclair J., Peskin A.* Development of nanotechnology and power systems concentrations for an on-line BSEET degree //. — American Society for Engineering Education, 2010.
 12. AR.js Documentation. — 2022. — URL: <https://ar-js-org.github.io/AR.js-Docs/>.
 13. *Arteaga J., Gravini-Donado M., Riva L.* Digital Technologies for Heritage Teaching: Trend Analysis in New Realities // International Journal of Emerging Technologies in Learning. — 2021. — Т. 16, № 21. — С. 132–148. — DOI: 10.3991/ijet.v16i21.25149.
 14. *Atmaca H., Terzi O.* Building a Web-Augmented Reality application for demonstration of kidney pathology for veterinary education // Polish Journal of Veterinary Sciences. — 2021. — Т. 24, № 3. — С. 345–350. — DOI: 10.24425/pjvs.2021.137671.
 15. Augmented Reality for Teaching Data Structures in Computer Science / H. Narman [та ін.] //. — Institute of Electrical, Electronics Engineers Inc., 2020. — DOI: 10.1109/GHTC46280.2020.9342932.
 16. Augmented reality software design for educational purposes / O. V. Syrovatskyi [та ін.] // CEUR Workshop Proceedings / за ред. S. O. Semerikov [та ін.]. — 2018. — Т. 2292. — С. 193–225. — ISSN 16130073. — URL: <http://ceur-ws.org/Vol-2292/paper20.pdf>.

17. Augmented Reality-Based Simulators as Discovery Learning Tools: An Empirical Study / M.-B. Ibanez [та ін.] // IEEE Transactions on Education. — 2015. — Т. 58, № 3. — С. 208—213. — DOI: 10.1109/TE.2014.2379712.
18. *Bhavika K., Martin J., Ardit B.* Technology will never replace hands on surgical training in plastic surgery // Journal of Plastic, Reconstructive and Aesthetic Surgery. — 2022. — Т. 75, № 1. — С. 439—488. — DOI: 10.1016/j.bjps.2021.11.034.
19. BoxGeometry. — URL: <https://threejs.org/docs/index.html#api/en/geometries/BoxGeometry>.
20. Build a prototype of new e-Learning contents by using the AR technology / A. Toguchi [та ін.] //. Т. 1. — International Institute of Informatics, Systemics, IIS, 2011. — С. 261—264.
21. *Centre for Science and Technology Studies, Leiden University, The Netherlands.* VOSviewer - Visualizing scientific landscapes. — 2022. — URL: <https://www.vosviewer.com/>.
22. *Cheng Y.* Teaching Machine Learning for K-12 with Teachable Machine. — 2020. — URL: <https://medium.com/readyai-org/teachable-machine-course-eb925c2c370a>.
23. Civil war battlefield experience: Historical event simulation using augmented reality technology / V. T. Nguyen [та ін.] //. — Institute of Electrical, Electronics Engineers Inc., 2019. — С. 294—297. — DOI: 10.1109/AIVR46125.2019.00068.
24. *Cortés Rodríguez F., Dal Peraro M., Abriata L.* Online tools to easily build virtual molecular models for display in augmented and virtual reality on the web // Journal of Molecular Graphics and Modelling. — 2022. — Т. 114. — DOI: 10.1016/j.jmgm.2022.108164.
25. *Coughlin T.* Impact of COVID-19 on the Consumer Electronics Market // IEEE Consumer Electronics Magazine. — 2021. — Т. 10, № 1. — С. 58—59. — DOI: 10.1109/MCE.2020.3016753.

26. Coupling AR with Object Detection Neural Networks for End-User Engagement / T. Katika [та ін.] // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) / за ред. G. Zachmann [та ін.]. — 2022. — Т. 13484 LNCS. — С. 135—145. — DOI: 10.1007/978-3-031-16234-3_8.
27. *Dar S.* Mobile library initiatives: a new way to revitalize the academic library settings // Library Hi Tech News. — 2019. — Т. 36, № 5. — С. 15—21. — DOI: 10.1108/LHTN-05-2019-0032.
28. *Demir I.* Interactive web-based hydrological simulation system as an education platform // Т. 2 / за ред. A. E. Rizzoli, N. W. T. Quinn, D. P. Ames. — International Environmental Modelling, Software Society, 2014. — С. 910—912.
29. Development of an augmented reality game to teach abstract concepts in food chemistry / P. Crandall [та ін.] // Journal of Food Science Education. — 2015. — Т. 14, № 1. — С. 18—23. — DOI: 10.1111/1541-4329.12048.
30. Development of career guidance quests using WebAR / D. S. Shepiliev [та ін.] // Journal of Physics: Conference Series. — 2021. — Бер. — Т. 1840, № 1. — С. 012028. — DOI: 10.1088/1742-6596/1840/1/012028. — URL: <https://doi.org/10.1088/1742-6596/1840/1/012028>.
31. Development of new e-Learning contents for improvement of laboratory courses by using the AR technology / A. Toguchi [та ін.] // International Institute of Informatics, Systemics, IIS, 2012. — С. 189—193.
32. Digital technologies to improve the precision of paediatric growth disorder diagnosis and management / L. Dunkel [та ін.] // Growth Hormone and IGF Research. — 2021. — Т. 59. — DOI: 10.1016/j.ghir.2021.101408.
33. Digital undergraduate education in dentistry: A systematic review / N. Zitzmann [та ін.] // International Journal of Environmental Research and Public Health. — 2020. — Т. 17, № 9. — DOI: 10.3390/ijerph17093269.

34. Distance learning in cardiology: The use of multimedia clinical diagnostic tasks / S. Karas [та иһ.] // Russian Journal of Cardiology. — 2020. — Т. 25, № 10. — С. 187—194. — DOI: 10.15829/1560-4071-2020-4116.
35. Effectiveness of Using Augmented Reality for Training in the Medical Professions: Meta-analysis / Y. Baashar [та иһ.] // JMIR Serious Games. — 2022. — Т. 10, № 3. — DOI: 10.2196/32715.
36. Effects of Virtual Reality Simulation on Worker Emergency Evacuation of Neonates / S. Farra [та иһ.] // Disaster Medicine and Public Health Preparedness. — 2019. — Т. 13, № 2. — С. 301—308. — DOI: 10.1017/dmp.2018.58.
37. Ehealth technologies to support nutrition and physical activity behaviors in diabetes self-management / M. Rollo [та иһ.] // Diabetes, Metabolic Syndrome and Obesity: Targets and Therapy. — 2016. — Т. 9. — С. 381—390. — DOI: 10.2147/DMSO.S95247.
38. Enhancing Visualisation of Anatomical Presentation and Education Using Marker-based Augmented Reality Technology on Web-based Platform / I. Wang [та иһ.] //. — Institute of Electrical, Electronics Engineers Inc., 2019. — DOI: 10.1109/AVSS.2018.8639147.
39. Enhancing vocational training in the post-covid era through mobile mixed reality / E. Smith [та иһ.] // Sustainability (Switzerland). — 2021. — Т. 13, № 11. — DOI: 10.3390/su13116144.
40. *Erçağ E., Yasakcı A.* The Perception Scale for the 7E Model-Based Augmented Reality Enriched Computer Course (7EMAGBAÖ): Validity and Reliability Study // Sustainability (Switzerland). — 2022. — Т. 14, № 19. — DOI: 10.3390/su141912037.
41. Evaluation of Surgical Improvement of Clinical Knowledge Ops (SICKO), an Interactive Training Platform / D. Nemirovsky [та иһ.] // Journal of Digital Imaging. — 2021. — Т. 34, № 4. — С. 1067—1071. — DOI: 10.1007/s10278-021-00482-x.
42. Experiences of web-based extended reality technologies for physics education / R. Zatarain-Cabada [та иһ.] // Computer Applications in Engineering Education. — 2022. — DOI: 10.1002/cae.22571.

43. Experimenting with electromagnetism using augmented reality: Impact on flow student experience and educational effectiveness / M. Ibáñez [та ил.] // Computers and Education. — 2014. — Т. 71. — С. 1—13. — DOI: 10.1016/j.compedu.2013.09.004.
44. Face Landmarks Detection. — 2022. — URL: <https://github.com/tensorflow/tfjs-models/tree/master/face-landmarks-detection>.
45. FI-AR learning: a web-based platform for augmented reality educational content / I. Coma-Tatay [та ил.] // Multimedia Tools and Applications. — 2019. — Т. 78, № 5. — С. 6093—6118. — DOI: 10.1007/s11042-018-6395-5.
46. FisiCArtivo: Design of a learning tool for physics education using web-based XR technology / B. Cardenas-Sainz [та ил.] //. — Institute of Electrical, Electronics Engineers Inc., 2022. — DOI: 10.1109/ENC56672.2022.9882930.
47. *Al-Gharaibeh J., Jeffery C.* Portable non-player character tutors with quest activities //. — 2010. — С. 253—254. — DOI: 10.1109/VR.2010.5444779.
48. *Google LLC.* MediaPipe Face Mesh. — URL: https://google.github.io/mediapipe/solutions/face_mesh.
49. *Gordon N., Brayshaw M., Aljaber T.* Heuristic evaluation for serious immersive games and M-instruction // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) / за ред. P. Zaphiris, A. Ioannou. — 2016. — Т. 9753. — С. 310—319. — DOI: 10.1007/978-3-319-39483-1_29.
50. *Hai-Jew S.* Adult Coloring Books as Emotional Salve/Stress Relief, Tactual-Visual Learning: An Analysis from Mass-Scale Social Imagery. — Nova Science Publishers, Inc., 2022. — С. 171—186.
51. Hand Pose Detection. — 2022. — URL: <https://github.com/tensorflow/tfjs-models/tree/master/hand-pose-detection>.

52. *Harun, Tuli N., Mantri A.* Experience Fleming's rule in electromagnetism using augmented reality: Analyzing impact on students learning // *T. 172 / за ред. R. Jagadeesh Kannan, L. Jeganathan, S. V. Nagaraj.* — Elsevier B.V., 2020. — С. 660—668. — DOI: 10.1016/j.procs.2020.05.086.
53. How to Improve Medical Simulation Training: A New Methodology Based on Ergonomic Evaluation / *A. Brunzini [та ін.] // Advances in Intelligent Systems and Computing / за ред. S. Nazir, W. Karwowski, T. Ahram.* — 2020. — Т. 963. — С. 145—155. — DOI: 10.1007/978-3-030-20135-7_14.
54. *Huang L.* Chemistry Apps on Smartphones and Tablets. — wiley, 2015. — С. 621—650. — DOI: 10.1002/9783527679300.ch25.
55. *Huang T.-C.* Seeing creativity in an augmented experiential learning environment // *Universal Access in the Information Society.* — 2019. — Т. 18, № 2. — С. 301—313. — DOI: 10.1007/s10209-017-0592-2.
56. Human factors and qualitative pedagogical evaluation of a mobile augmented reality system for science education used by learners with physical disabilities / *T. Arvanitis [та ін.] // Personal and Ubiquitous Computing.* — 2009. — Т. 13, № 3. — С. 243—250. — DOI: 10.1007/s00779-007-0187-7.
57. *Ibáñez M. B., Peláez J., Kloos C. D.* Using an augmented reality geolocalized quiz game as an incentive to overcome academic procrastination // *Advances in Intelligent Systems and Computing / за ред. M. E. Auer, T. Tsiatsos.* — 2019. — Т. 909. — С. 175—184. — DOI: 10.1007/978-3-030-11434-3_21.
58. 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 / *M. Ibáñez [та ін.] // Computers and Education.* — 2020. — Т. 145. — DOI: 10.1016/j.compedu.2019.103734.
59. Implementation of augmented reality in "3.0 learning" methodology: Case studies with students of architecture degree / *E. Redondo [та ін.]*. —

- IGI Global, 2013. — C. 391—413. — DOI: 10.4018/978-1-4666-2851-9.ch019.
60. 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 / A. Vahabzadeh [та ит.] // JMIR Mental Health. — 2018. — Т. 5, № 2. — DOI: 10.2196/mental.9631.
 61. Installation. — URL: <https://threejs.org/docs/index.html#manual/en/introduction/Installation>.
 62. Integrating Robo-PEM with AR Application for Introducing Fuel Cell Implementation / S. Sendari [та ит.] //. — Institute of Electrical, Electronics Engineers Inc., 2021. — DOI: 10.1109/ICEEIE52663.2021.9616683.
 63. International Conference on Applied and Practical Sciences, ICAPS 2021 //. Т. 1860. — IOP Publishing Ltd, 2021.
 64. *Jara C., Candelas F., Torres F.* Internet virtual and remote control interface for robotics education. — Nova Science Publishers, Inc., 2009. — C. 136—154.
 65. *Kazanidis I., Pellas N., Christopoulos A.* A learning analytics conceptual framework for augmented reality-supported educational case studies // Multimodal Technologies and Interaction. — 2021. — Т. 5, № 3. — DOI: 10.3390/mti5030009.
 66. *Klavins A.* 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/>.
 67. *Le H., Nguyen M.* An Online Platform for Enhancing Learning Experiences with Web-Based Augmented Reality and Pictorial Bar Code // Springer Series on Cultural Computing. — 2020. — C. 45—57. — DOI: 10.1007/978-3-030-42156-4_3.
 68. *Lou D.* Two fast prototypes of web-based augmented reality enhancement for books // Library Hi Tech News. — 2019. — Т. 36, № 10. — C. 19—24. — DOI: 10.1108/LHTN-08-2019-0057.

69. *Lytridis C., Tsinakos A., Kazanidis I.* ARTutor—An augmented reality platform for interactive distance learning // Education Sciences. — 2018. — Т. 8, № 1. — DOI: 10.3390/educsci8010006.
70. *Marín R., Sanz P. J.* The human-machine interaction through the UJI Telerobotic Training System // / за ред. М. Н. Hamza. — 2003. — С. 47—52.
71. *Marín R., Sanz P. J., Del Pobil A. P.* The UJI Online Robot: An Education and Training Experience // Autonomous Robots. — 2003. — Т. 15, № 3. — С. 283—297. — DOI: 10.1023/A:1026220621431.
72. *MDN contributors.* WebGL: 2D and 3D graphics for the web. — 2022. — URL: https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API.
73. Mobile learning approach as a supplementary approach in the organization of the studying process in educational institutions / S. L. Proskura [та ін.] // Т. 2732 / за ред. О. Sokolov [та ін.]. — CEUR-WS, 2020. — С. 650—664.
74. MobileNet. — 2022. — URL: <https://github.com/tensorflow/tfjs-models/tree/master/mobilenet>.
75. Need Finding for an Embodied Coding Platform: Educators' Practices and Perspectives / Т. Sharkey [та ін.] // Т. 1 / за ред. М. Cukurova [та ін.]. — Science, Technology Publications, Lda, 2022. — С. 216—227. — DOI: 10.5220/0011000200003182.
76. Next generation of chemistry and biochemistry conference posters: Animation, augmented reality, visitor statistics, and visitors' attention / E. Faridi [та ін.] // Biochemistry and Molecular Biology Education. — 2021. — Т. 49, № 4. — С. 619—624. — DOI: 10.1002/bmb.21520.
77. ngrok - Online in One Line. — 2022. — URL: <https://ngrok.com/>.
78. *Nguyen V., Jung K., Dang T.* BlocklyAR: A visual programming interface for creating augmented reality experiences // Electronics (Switzerland). — 2020. — Т. 9, № 8. — С. 1—20. — DOI: 10.3390/electronics9081205.

79. *Nordin N., Nordin N., Omar W.* REV-OPOLY: A Study on Educational Board Game with Webbased Augmented Reality // Asian Journal of University Education. — 2022. — Т. 18, № 1. — С. 81–90. — DOI: 10.24191/ajue.v18i1.17172.
80. Palmitoar: The last battle of the U.S. Civil war reenacted using augmented reality / K. Jung [та ін.] // ISPRS International Journal of Geo-Information. — 2020. — Т. 9, № 2. — DOI: 10.3390/ijgi9020075.
81. Pose Detection in the Browser: PoseNet Model. — 2021. — URL: <https://github.com/tensorflow/tfjs-models/tree/master/posenet>.
82. Procedia Computer Science // Т. 135 / за ред. А. У. Meiliana [та ін.]. — Elsevier B.V., 2018.
83. Proceedings - 2014 International Conference on Teaching and Learning in Computing and Engineering, LATICE 2014 // — IEEE Computer Society, 2014.
84. Proceedings of 2015 International Conference on Interactive Mobile Communication Technologies and Learning, IMCL 2015 // — Institute of Electrical, Electronics Engineers Inc., 2015.
85. *Proskura S. L., Lytvynova S. H.* The approaches to Web-based education of computer science bachelors in higher education institutions // Т. 2643 / за ред. А. Е. Kiv, М. Р. Shyshkina. — CEUR-WS, 2020. — С. 609–625.
86. Rapid Curricular Innovations During COVID-19 Clinical Suspension: Maintaining Student Engagement with Simulation Experiences / H. Bradford [та ін.] // Journal of Midwifery and Women's Health. — 2021. — Т. 66, № 3. — С. 366–371. — DOI: 10.1111/jmwh.13246.
87. Rapid deployment of smartphone-based augmented reality tools for field and online education in structural biology / Т. Hoog [та ін.] // Biochemistry and Molecular Biology Education. — 2020. — Т. 48, № 5. — С. 448–451. — DOI: 10.1002/bmb.21396.
88. *ReadyAI.* Teachable Machine Lesson Plan. — 2020. — URL: <https://edu.readyai.org/courses/teachable-machine/lessons/teachable-machine-lesson-plan/>.

89. Reassessing Undergraduate Polymer Chemistry Laboratory Experiments for Virtual Learning Environments / M. Karayilan [та ін.] // Journal of Chemical Education. — 2021. — DOI: 10.1021/acs.jchemed.1c01259.
90. *Ruhlemann D.* Facebook AR Studio – How to: Use Tap Gestures to Change Effect Materials. — 2018. — URL: <https://www.tallan.com/blog/2018/04/24/facebook-ar-studio-how-to-use-tap-gestures-to-change-effect-materials/>.
91. *Samat C., Chaijaroen S.* Design and Development of Constructivist Augmented Reality (AR) Book Enhancing Analytical Thinking in Computer Classroom // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) / за ред. L. Ronningsbakk [та ін.]. — 2019. — Т. 11937 LNCS. — С. 175–183. — DOI: 10.1007/978-3-030-35343-8_19.
92. *Semerikov S. O., Mintii M. M., Mintii I. S.* Review of the course “Development of Virtual and Augmented Reality Software” for STEM teachers: implementation results and improvement potentials // Proceedings of the 4th International Workshop on Augmented Reality in Education (AREdu 2021), Kryvyi Rih, Ukraine, May 11, 2021. Т. 2898 / за ред. S. H. Lytvynova, S. O. Semerikov. — CEUR-WS.org, 2021. — С. 159–177. — (CEUR Workshop Proceedings). — URL: <http://ceur-ws.org/Vol-2898/paper09.pdf>.
93. *Sepasgozar S.* Digital twin and web-based virtual gaming technologies for online education: A case of construction management and engineering // Applied Sciences (Switzerland). — 2020. — Т. 10, № 13. — DOI: 10.3390/app10134678.
94. Setup - ngrok. — 2022. — URL: <https://dashboard.ngrok.com/get-started/your-auth-token>.
95. *Sharp K., McCorvie M., Wagner M.* Sharing Hidden Histories: The XRchaeology at Miller Grove, a Free African American Community in Southern Illinois // Journal of African Diaspora Archaeology and Heritage. — 2021. — DOI: 10.1080/21619441.2021.1902706.

96. Small and flexible web based framework for teaching QR and AR mobile learning application development / J. Ferguson [та ін.] // . — Institute of Electrical, Electronics Engineers Inc., 2015. — С. 383—385. — DOI: 10.1109/IMCTL.2015.7359624.
97. *Spasova N., Ivanova M.* Towards augmented reality technology in cad/cam systems and engineering education // / за ред. I. Roceanu. — National Defence University - Carol I Printing House, 2020. — С. 496—503. — DOI: 10.12753/2066-026X-20-151.
98. Speech Command Recognizer. — 2022. — URL: <https://github.com/tensorflow/tfjs-models/tree/master/speech-commands>.
99. Studies in Health Technology and Informatics // Studies in Health Technology and Informatics / за ред. L. Fellander-Tsai [та ін.]. — 2016. — Т. 220.
100. Teachable Machine. — URL: <https://teachablemachine.withgoogle.com/>.
101. TEaching Earth Systems Beyond the Classroom: Developing a Mixed Reality (XR) Sandbox / D. Tennakoon [та ін.] // . — American Society for Engineering Education, 2022.
102. TensorFlow. — URL: <https://www.tensorflow.org/>.
103. TensorFlow Hub. — URL: <https://tfhub.dev/>.
104. TensorFlow.js | Machine Learning for JavaScript Developers. — 2022. — URL: <https://www.tensorflow.org/js>.
105. TensorFlow.js models. — URL: <https://www.tensorflow.org/js/models>.
106. *Thabvithorn C., Samat C.* Development of Web-Based Learning with Augmented Reality (AR) to Promote Analytical Thinking on Computational Thinking for High School // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) / за ред. Y.-M. Huang [та ін.]. — 2022. — Т. 13449 LNCS. — С. 125—133. — DOI: 10.1007/978-3-031-15273-3_14.

107. The Face Mask Template in Adobe Photoshop. — URL: <https://sparkar.facebook.com/ar-studio/learn/articles/creating-and-prepping-assets/the-face-mask-template-in-Adobe/>.
108. Three.js – JavaScript 3D Library. — 2022. — URL: <https://threejs.org/>.
109. three.js editor. — URL: <https://threejs.org/editor/>.
110. *TrackJS LLC*. Remote JavaScript Debugger - RemoteJS. — 2022. — URL: <https://remotejs.com/>.
111. *Tuli N., Mantri A., Sharma S.* Impact of augmented reality tabletop learning environment on learning and motivation of kindergarten kids // T. 2357 / за ред. А. Upmanyu [та ин.]. — American Institute of Physics Inc., 2022. — DOI: 10.1063/5.0080600.
112. *Turner F., Welch I.* The mixed reality toolkit as the next step in the mass customization co-design experience // International Journal of Industrial Engineering and Management. — 2019. — Т. 10, № 2. — С. 191–199. — DOI: 10.24867/IJIEM-2019-2-239.
113. *UNPKG*. UNPKG. — 2022. — URL: <https://unpkg.com/>.
114. Using extended reality (XR) for medical training and real-time clinical support during deep space missions / B. Burian [та ин.] // Applied Ergonomics. — 2023. — Т. 106. — DOI: 10.1016/j.apergo.2022.103902.
115. Using the Meta Spark Studio Face Reference Assets. — URL: <https://sparkar.facebook.com/ar-studio/learn/articles/people-tracking/face-reference-assets#optimization>.
116. Versatile Mixed Reality Educational Spaces; A Medical Education Implementation Case / P. E. Antoniou [та ин.] // / за ред. N. Georgalas [та ин.]. — Institute of Electrical, Electronics Engineers Inc., 2017. — С. 132–137. — DOI: 10.1109/IUCC-CSS.2016.026.
117. Versatile mixed reality medical educational spaces; requirement analysis from expert users / P. E. Antoniou [та ин.] // Personal and Ubiquitous Computing. — 2017. — Т. 21, № 6. — С. 1015–1024. — DOI: 10.1007/s00779-017-1074-5.

118. *Villarán D., Ibáñez M. B., Kloos C. D.* Augmented reality-based simulations embedded in problem based learning courses // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) / за ред. J. Konert [та ін.]. — 2015. — Т. 9307. — С. 540—543. — DOI: 10.1007/978-3-319-24258-3_55.
119. Web Server for Chrome. — 2021. — URL: <https://chrome.google.com/webstore/detail/web-server-for-chrome/ofhbbkphhbklhfoeikjpcbhmlcggigb?hl=en>.
120. WebAR development tools: An overview / D. S. Shepiliev [та ін.] // CEUR Workshop Proceedings. — 2020. — Т. 2832. — С. 84—93. — URL: <http://ceur-ws.org/Vol-2832/paper12.pdf>.
121. WebAR development tools: comparative analysis / V. V. Tkachuk [та ін.] // Physical and Mathematical Education. — 2020. — 2(24). — DOI: 10.31110/2413-1571-2020-024-2-021. — URL: <https://doi.org/10.31110%2F2413-1571-2020-024-2-021>.
122. *WebART*: Web-based Augmented Reality Learning Resources Authoring Tool and Its User Experience Study among Teachers / E. Liu [та ін.] // IEEE Transactions on Learning Technologies. — 2022. — С. 1—14. — DOI: 10.1109/TLT.2022.3214854.
123. *Wongchiranuwat S., Samat C.* Synthesis of theoretical framework for augmented Reality learning environment to promote creative thinking on topic implementation of graphic design for grade 9 students // / за ред. S. L. Wong [та ін.]. — Asia-Pacific Society for Computers in Education, 2016. — С. 639—641.
124. Work-in-Progress-Development of a Virtual Reality Learning Environment: VR Baby / G. Ryan [та ін.] // / за ред. D. Economou [та ін.]. — Institute of Electrical, Electronics Engineers Inc., 2020. — С. 312—315. — DOI: 10.23919/iLRN47897.2020.9155203.
125. *Yang S., Mei B., Yue X.* Mobile Augmented Reality Assisted Chemical Education: Insights from Elements 4D // Journal of Chemical Education. — 2018. — Т. 95, № 6. — С. 1060—1062. — DOI: 10.1021/acs.jchemed.8b00017.

126. *Yuen H.* HiuKim Yuen. — 2022. — URL: <https://www.youtube.com/channel/UC-JyA1Z1-p0wgxj5WEX56wg/featured>.
127. *Yuen H.* MindAR. — 2022. — URL: <https://hiukim.github.io/mind-ar-js-doc/>.

Додаток А

Вихідний код користувацької бібліотеки loader.js

```
import {GLTFLoader} from
  "https://unpkg.com/three/examples/jsm/loaders/GLTFLoader.js";
import * as THREE from
  "https://unpkg.com/three/build/three.module.js";

export const loadGLTF = (path) => {
  return new Promise((resolve, reject) => {
    const loader = new GLTFLoader();
    loader.load(path, (gltf) => {
      resolve(gltf);
    });
  });
}

export const loadAudio = (path) => {
  return new Promise((resolve, reject) => {
    const loader = new THREE.AudioLoader();
    loader.load(path, (buffer) => {
      resolve(buffer);
    });
  });
}

export const loadVideo = (path) => {
  return new Promise((resolve, reject) => {
    const video = document.createElement("video");
    video.addEventListener('loadeddata', () => {
      video.setAttribute('playsinline', '');
      resolve(video);
    });
  });
}
```

```

    video.src = path;
  });
}

export const loadTexture = (path) => {
  return new Promise((resolve, reject) => {
    const loader = new THREE.TextureLoader();
    loader.load(path, (texture) => {
      resolve(texture);
    });
  });
}

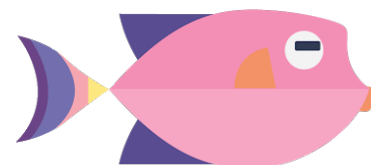
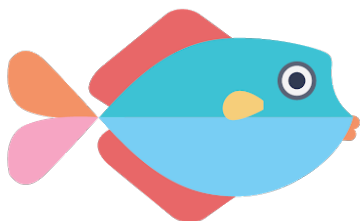
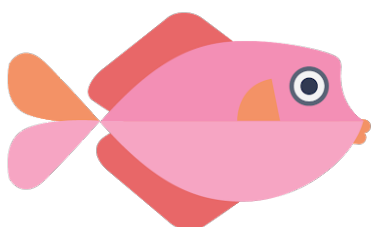
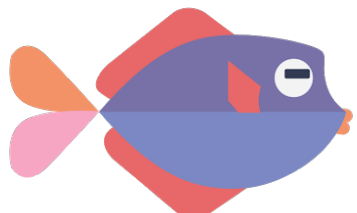
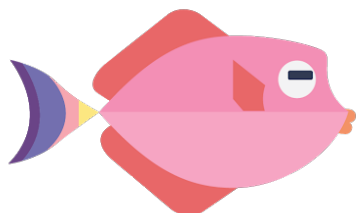
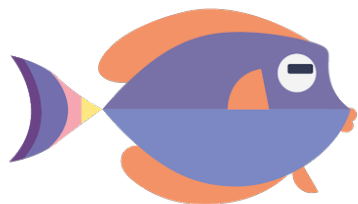
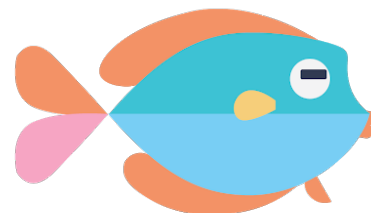
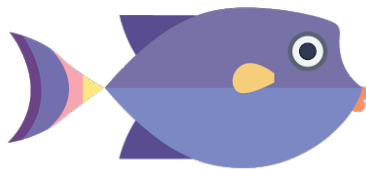
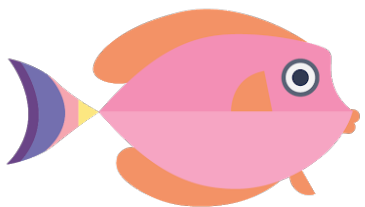
export const loadTextures = (paths) => {
  const loader = new THREE.TextureLoader();
  const promises = [];
  for (let i = 0; i < paths.length; i++) {
    promises.push(new Promise((resolve, reject) => {
      loader.load(paths[i], (texture) => {
        resolve(texture);
      });
    }));
  }
  return Promise.all(promises);
}

```

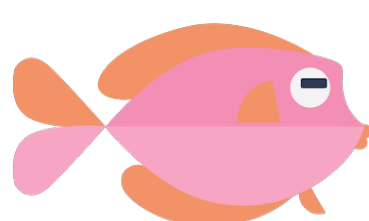
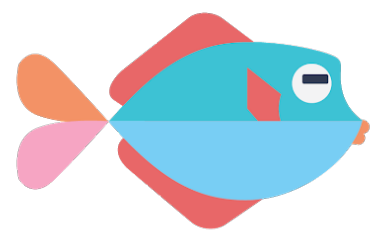
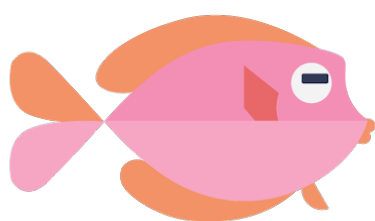
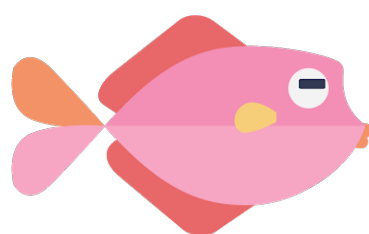
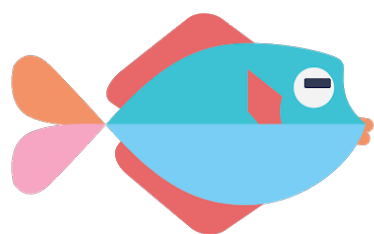
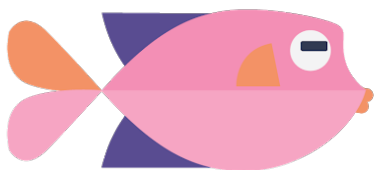
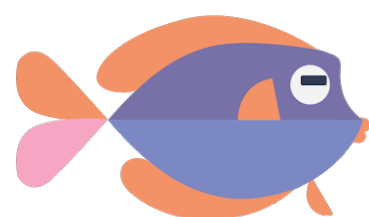
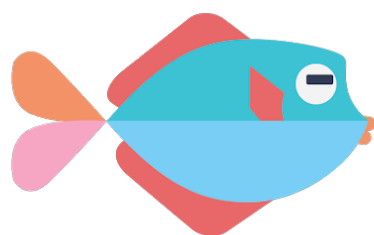
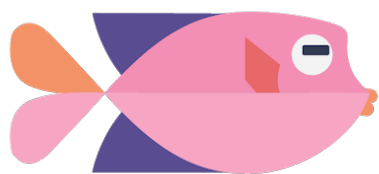
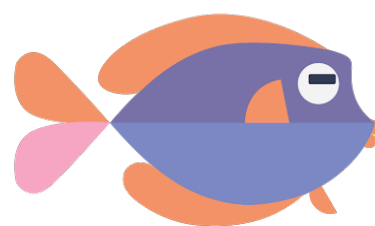
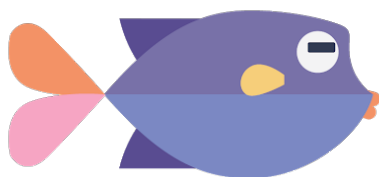
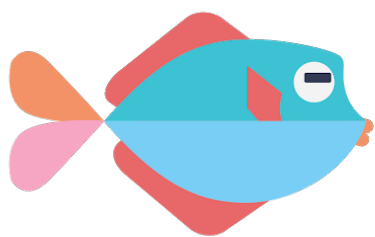
Додаток Б

Роздатковий матеріал “Класифікація риб”

Отруйні риби

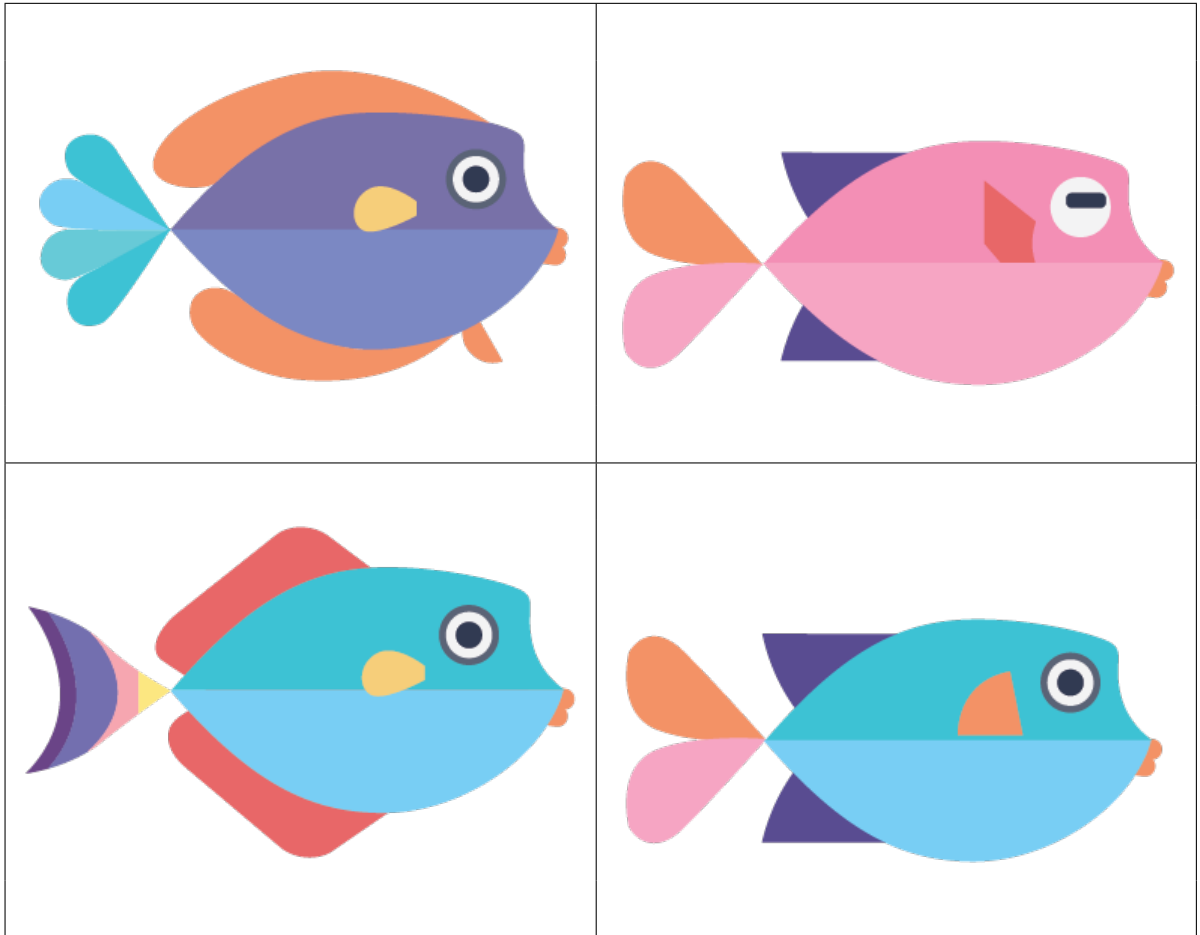


Неотруйні риби



ТЕСТОВІ ДАНІ

Ґрунтуючись на Ваших гіпотезах, визначте, ці риби отруйні чи неотруйні?



Додаток В

Робочий аркуш “Дерево рішень”

Словник

- **Машинне навчання:** машини демонструють здатність навчатися без явного програмування.
- **Дерево рішень:** набір правил, що їх ми можемо використати для поділу даних на категорії (виглядає як дерево).

Які фізичні загрози важливі при визначенні, чи є риба отруйною?

Скористайтесь цим місцем, щоб намалювати власне дерево рішень.

Поясніть, як Ви дійшли таких висновків?