

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Сумський державний університет**

**Факультет електроніки та інформаційних технологій**

**Кафедра комп'ютерних наук**

«До захисту допущено»

В.о. завідувача кафедри

**Ігор ШЕЛЕХОВ**

\_\_\_\_\_

(підпис)

\_\_\_\_\_

**КВАЛІФІКАЦІЙНА РОБОТА**

**на здобуття освітнього ступеня магістр**

зі спеціальності 122 - Комп'ютерних наук,

освітньо-професійної програми «Інформатика»

на тему: «Інтелектуальна технологія запобігання витоку даних через  
мультимедійний контент»

здобувача групи ІН.м - 25 Галевича Сергія Сергійовича

Кваліфікаційна робота містить результати власних досліджень.  
Використання ідей, результатів і текстів інших авторів мають посилання на  
відповідне джерело.

Сергій Галевич

\_\_\_\_\_

(підпис)

Керівник,

кандидат технічних наук

**Борис КУЗІКОВ**

\_\_\_\_\_

(підпис)

**Суми – 2023**

**Сумський державний університет**  
Факультет електроніки та інформаційних технологій  
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

\_\_\_\_\_ (підпис)

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ**

**на здобуття освітнього ступеня магістра**

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»  
здобувача групи ІН.м-25 Галевича Сергія Сергійовича

1. Тема роботи: «Інтелектуальна технологія запобігання витоку даних через мультимедійний контент»

затверджую наказом по СумДУ від «06» грудня 2023 року № 1412-VI

2. Термін здачі здобувачем кваліфікаційної роботи до 18 грудня 2023 року

3. Вхідні дані до кваліфікаційної роботи \_\_\_\_\_

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1) Аналіз проблеми предметної області, постановка й формування завдань дослідження.

2) Огляд алгоритмів комп'ютерного зору для виявлення об'єктів. 3) Розробка інтелектуальної системи запобігання витоку медіаконтенту. 4) Аналіз результатів.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) \_\_\_\_\_

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

Завдання прийняв до виконання \_\_\_\_\_  
(підпис)

Керівник \_\_\_\_\_  
(підпис)

**КАЛЕНДАРНИЙ ПЛАН**

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблеми предметної області, постановка й формування завдань дослідження</i>		
2	<i>Огляд алгоритмів комп'ютерного зору, що використовуються для виявлення об'єктів</i>		
3	<i>Розробка інтелектуальної системи запобігання витоку медіаконтенту</i>		
4	<i>Аналіз отриманих результатів</i>		
5	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>		

Здобувач вищої освіти \_\_\_\_\_  
(підпис)

Керівник \_\_\_\_\_  
(підпис)

## АНОТАЦІЯ

**Записка:** 59 стр., 20 рис., 1 додаток, 25 використаних джерел.

**Обґрунтування актуальності теми роботи** – Тема кваліфікаційної роботи є актуальною, оскільки присвячена розв’язанню важливої практичної задачі запобігання витоку чутливої інформації шляхом її розпізнавання та маскування з використанням моделей та методів комп’ютерного зору.

**Об’єкт дослідження** — розпізнавання образів.

**Мета роботи** — розробка інформаційної технології запобігання витоку даних через мультимедійний контент.

**Методи дослідження** — моделі та методи комп’ютерного зору, навчання із учителем.

**Результати** — розроблено інформаційну технологію, що дозволяє виявляти на фото-зображеннях та наносити непрозору маску на вразливі об’єкти, запобігаючи можливому витоку інформації. Реалізовано інформаційну систему, як частину означеної технології. Проведено тестування на матеріалах із відкритих джерел. Виділено сильні та слабкі сторони та шляхи подальшого розвитку інформаційної технології.

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ, ШТУЧНИЙ ІНТЕЛЕКТ, КОМП’ЮТЕРНИЙ  
ЗІР, РОЗПІЗНОВАННЯ ОБРАЗІВ, TENSORFLOW.JS, VUE.JS, YOLO

## ЗМІСТ

ВСТУП.....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	7
1.1 Дослідження актуальності проблеми .....	7
1.2 Взаємодія OSINT з штучним інтелектом .....	15
1.3 Постановка задачі для реалізації.....	19
2 НАВЧАННЯ МОДЕЛІ .....	21
2.1 Огляд алгоритмів комп'ютерного зору для виявлення об'єктів.....	21
2.2 Пошук, підготовка та анотація зображень .....	25
2.3 Навчання моделі .....	29
3 РОЗРОБКА ДОДАТКУ .....	36
3.1 Вибір фреймворків .....	36
3.2 Опис програмної реалізації.....	38
3.3 Аналіз результатів .....	41
ВИСНОВКИ.....	45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	46
ДОДАТОК А .....	49

## ВСТУП

**Актуальність.** Сучасне суспільство щоденно генерує велику кількість медіа контенту, публікуючи його у відкритому доступі. Відео-камера тепер є у кожному телефоні. Медіа контент несе в собі потенційні загрози, зокрема, можливість розповсюдження приватної та чутливої інформації без належного контролю. В умовах війни в Україні, це питання набуває особливої важливості. Контент, що містить зображення військових об'єктів та приватний спосіб життя воїнів, може стати об'єктом уваги ворожих сил. Недбале використання соціальних мереж може призвести до витоку конфіденційної інформації, що, в свою чергу, викликає потенційні загрози безпеці.

Збереження конфіденційності та захист інформації стають ключовими завданнями в умовах збройного протистояння, а розпізнавання чутливого контенту на фотозображеннях стає важливим завданням у сфері інформаційної безпеки. Тому розробка інформаційної технології, яка дозволяє автоматично розпізнавати такий контент та захищати його шляхом нанесення маски є актуальним завданням.

**Об'єкт дослідження** – розпізнавання образів.

**Предмет дослідження** – застосування моделей та методів комп'ютерного зору для запобігання витоку інформації через фото-матеріали у цифровій формі.

**Гіпотеза.** Автоматичне розпізнавання та маскуванню чутливої інформації на фото зображеннях дозволить знизити ризик витоку інформації та підвищити рівень обізнаності користувачів у сфері інформаційної гігієни.

**Новизна.** Розроблено інформаційну технологію, що дозволяє запобігати витоку потенційно чутливої інформації у автоматичному режимі. Програмні засоби обробляють інформацію з боку клієнта без використання серверу, запобігаючи її витоку через канали зв'язку чи злам серверу. Розроблена технологія може бути використана як засіб підвищення обізнаності про інформанту гігієну.

**Структура.** Кваліфікаційна дипломна робота складається з вступу, аналізу предметної області, постановки задачі, вибору та огляду алгоритмів комп'ютерного зору, пошуку та підготовки зображень для навчання моделі штучного інтелекту, вибору фреймворків для розробки веб-додатку, аналізу результатів, висновку, списків використаних джерел у роботі та додатку.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Дослідження актуальності проблеми

В епоху, коли цифровізація невпинно розширює свої кордони, наш світ зазнає неймовірних змін. Цифровізація - це процес переведення інформації в цифровий формат. Час, коли інформація була відносно обмеженою та зберігалася у фізичних архівах, відійшов у минуле. Тепер величезні обсяги даних, які колись були доступні лише для вузького кола спеціалістів, лежать на поверхні для всіх, хто має доступ до інтернету.

Цифровізація не може існувати без динамічного розвитку інновацій — її необхідної умови. У процесі розвитку цивілізації по відношенню до цієї епохи з'являється прогрес визначення четвертої промислової революції, як цифрової трансформації початку 21 століття. Це не раптове відкриття, а результат багатовікового розвитку людства, з яким нерозривно пов'язаний розвиток людських знань. При її створенні, обміні (поширенні), технічні засоби відіграють важливу роль. Їх багатовіковий і повільний розвиток прискорюється на зламі 20-21 століття.

Діяльність на основі цифрового контенту робить доступ до інформації більш динамічнішим, але має певний ефект витрат і створюються нові умови для інновацій; оцифрування та трансформації, стають одним із чинників інновацій. Після перетворення даних вони формуються для сумісності з різними цифровими пристроями та програмним забезпеченням. Цифрові дані легко зберігаються на різних електронних носіях: жорстких дисках, SSD, у хмарному сховищі, яке спрощує процес зберігання та управління. Цифровізація дозволяє легко передавати інформацію через мережу Інтернет або іншим способом, полегшуючи спільне використання даних та доступ до них.

У процесі розвитку хмарних технологій з'явилася можливість працювати з великими даними (big data), аналізуючи великі обсяги інформації.

До основних переваг процесу цифровізації провідні експерти відносять:

- автоматизацію процесів, які дозволяють швидко виконувати поточні завдання та оптимізувати продажі;
- можливість швидко перевірити популярні продукти компанії;
- впорядкування документації;
- удосконалення процедури обслуговування клієнтів;
- покращення комунікації;
- миттєве реагування на виклики;
- забезпечення безпеки даних.

Нижче наведено графік (рис 1.1), який ілюструє зростання ринку, починаючи з 2000 по 2023 рік у вигляді обсягу даних в ексабайтах (1 ексабайт = 1 000 000 терабайт). На цьому графіку також відображені ключові моменти, які сприяли цьому росту:

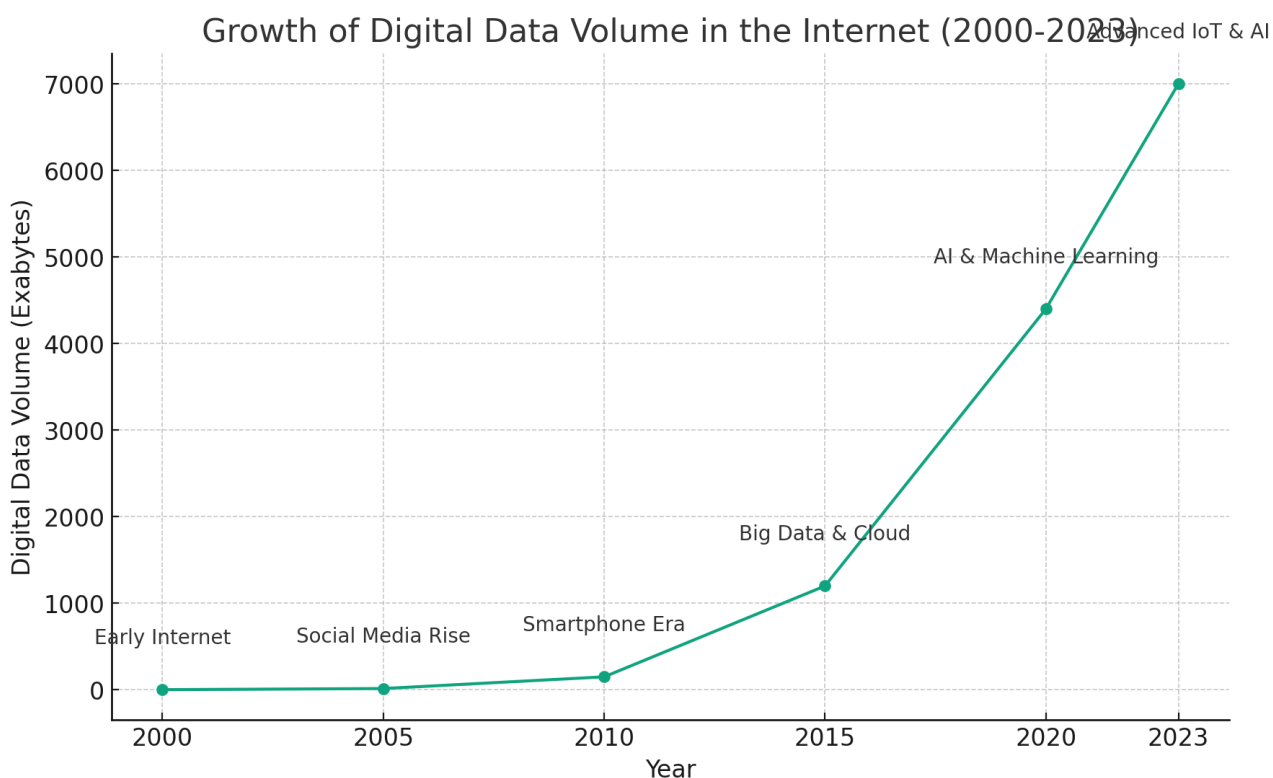


Рисунок 1.1 – Графік відображення ключових моментів, які сприяли цьому росту



1) 2000 рік – початок епохи Інтернету. Саме цей рік відзначається початком активного розвитку. Переважно інтернет використовувався для простих операцій, перетворених у цифру, зокрема: електронна пошта, пошук веб-сторінок, аналіз статистики тощо. Більшість операцій були досить простими, а обсяг генерованої інформації невеликий, порівняно із сучасними стандартами;

2) 2005 рік – період розвитку соціальних медіа та початок популярності таких платформ як Facebook, YouTube, Twitter. У цей період активно збільшилася кількість контенту, який генерувався користувачами.

3) 2010 рік — початок ери смартфонів. Це період, коли доступ до інтернету став значно простішим. А через соціальні медіа відбулося ще більше зростання кількості генерованих даних.

4) 2015 рік — час розвитку «великих» даних та хмарних технологій. Розвиток хмарних обчислень забезпечив потужні інструменти для зберігання, обробки та аналізу великих обсягів інформації. Технології почали активно використовуватися у різних сферах людської діяльності.

5) 2020 рік – час розвитку штучного інтелекту та машинного навчання. Прогрес стимулював процеси створення та обробки ще більших обсягів даних.

6) 2023 рік — розвинений інтернет та ШІ (IoT) став причиною того, що щодня більше пристроїв можна з'єднувати між собою та інтелектуальними системами, генеруючи великі обсяги інформації.

Усі ці етапи мали значний вплив на спосіб генерації інформації, що в свою чергу призвело до зростання кількості медіаконтенту. Завдяки розвитку цифрових технологій, створення медіаконтенту стало значно доступнішим. Сучасні цифрові пристрої, як-от смартфони з високоякісними камерами та легкодоступні програми для редагування, дозволяють кожному стати творцем контенту. Це спричинило величезний приплив контенту в інтернеті, особливо через платформи соціальних медіа та сервіси обміну відео.

Роль цифровізації у професійному медіа-виробництві також не можна недооцінювати. Вона спростила процеси зйомки, редагування та розповсюдження фільмів, телешоу та музики, роблячи їх більш ефективними та коштовними. Цифрові платформи для стрімінгу, такі як Netflix та Spotify, та соц мережі Instagram і YouTube та Twitter ( на даний момент назва X ) революціонізували спосіб доступу до медіаконтенту, пропонуючи безпрецедентний вибір та зручність для споживачів. Водночас, цифровізація сприяє глобалізації медіаконтенту, забезпечуючи легке перетворення та адаптацію контенту для міжнародної аудиторії.

Отже, цифровізація не лише збільшує кількість доступного медіаконтенту, але й сприяє його різноманітності та доступності. Вона відкриває нові можливості для творців контенту, розширює межі традиційних медіа та перетворює спосіб, яким ми споживаємо медіа в нашому щоденному житті. В свою чергу, таке гігантське зростання обсягу публічної інформації та медіаконтенту, доступних кожному призводить до зростання попиту розвідки.

Інформаційна розвідка є комплексним процесом збору, аналізу та використання даних для отримання цінної інформації, яка може бути використана в різних сферах, від бізнесу до національної безпеки. Вона включає в себе збір інформації з різноманітних джерел, її систематичний аналіз та інтерпретацію з метою розуміння певних тенденцій, виявлення потенційних можливостей та загроз, а також для підтримки прийняття обґрунтованих рішень. Важливо розуміти, що інформаційна розвідка не обмежується лише збором даних; вона також включає в себе критичний аналіз цих даних. Це може включати оцінку надійності джерела, контекстуалізацію інформації, визначення зв'язків між різними фрагментами даних та їх агрегацію для отримання повної картини.

Для прикладу, у бізнесі інформаційна розвідка може використовуватися для виявлення нових ринкових можливостей. Компанія може аналізувати дані

про споживацькі настрої, тенденції попиту, конкурентні стратегії та технологічні інновації, щоб розробити нові продукти або адаптувати свої маркетингові стратегії. Наприклад, компанія, яка виробляє спортивне взуття, може використовувати інформаційну розвідку для аналізу трендів у соціальних медіа та онлайн-оглядах, щоб зрозуміти споживацькі переваги та зміни в стилях життя, які можуть вплинути на попит.

У сфері національної безпеки, інформаційна розвідка використовується для оцінки загроз, зовнішньої та внутрішньої політики інших країн, а також для моніторингу потенційних кібератак. Наприклад, урядові агентства можуть аналізувати зв'язки між різними терористичними групами, відстежувати їх комунікації та фінансові транзакції, щоб запобігти атакам та захистити національну безпеку.

В епоху цифрових технологій інформаційна розвідка стає все більш важливою завдяки росту обсягів доступних даних та розвитку інструментів для їх аналізу. Великі дані (big data), машинне навчання та штучний інтелект відкривають нові можливості для глибшого аналізу та виявлення важливих інсайтів. Наприклад, з допомогою алгоритмів машинного навчання можна аналізувати великі обсяги даних з соціальних медіа, щоб визначити зміни в споживацьких настроях або відстежити розповсюдження дезінформації.

Окрім збору та аналізу даних, інформаційна розвідка також включає в себе заходи щодо захисту інформації. Це особливо важливо у світі, де кіберзагрози стають все більш розповсюдженими. Захист інформаційних систем від несанкціонованого доступу та кібератак є критично важливим для забезпечення цілісності та конфіденційності даних.

В сучасному світі, де інформація стає ключовим активом, інформаційна розвідка відіграє важливу роль у підтримці конкурентоспроможності, безпеки та інноваційності організацій та держав. Вона допомагає розуміти складність сучасного інформаційного середовища, виявляти приховані зв'язки та

використовувати цю інформацію для прийняття обґрунтованих та вчасних рішень.

Цифровізація значно впливає на розвиток інформаційної розвідки, змінюючи спосіб збору, аналізу та використання інформації. Завдяки зростанню обсягу доступних даних, організації та індивіди мають можливість отримати глибший і ширший доступ до інформації, що може бути використана для розуміння ринкових тенденцій, оцінки конкурентного середовища та виявлення нових можливостей та загроз. Соціальні медіа та інші онлайн-платформи стали неоціненними джерелами інсайтів, надаючи інформацію про споживацькі настрої, реакції на продукти та послуги, а також даючи змогу відстежувати діяльність конкурентів.

Розвиток аналітичних інструментів, таких як великі дані, машинне навчання та штучний інтелект, відкрив нові можливості для глибшого та більш точного аналізу великих обсягів інформації. Ці технології дозволяють компаніям та аналітикам краще розуміти ринкові тенденції та споживацьку поведінку. Крім того, зростаюча потреба в захисті інформаційних активів сприяла розвитку кіберрозвідки, яка включає моніторинг кіберпростору для виявлення потенційних загроз, таких як хакерські атаки та шпигунське програмне забезпечення.

Цифровізація також внесла зміни у процеси прийняття рішень у бізнесі та уряді. Завдяки доступу до різноманітних джерел інформації та вдосконалених аналітичних інструментів, рішення стають більш інформованими та заснованими на даних. Це дозволяє організаціям краще реагувати на зміни ринкових умов та ефективно адаптуватися до нових викликів.

Загалом, цифровізація та зростання медіаконтенту не тільки сприяють розвитку інформаційної розвідки, але й роблять її більш ефективною, точною та важливою для різних галузей. Вона відкриває нові можливості для аналізу, стратегічного планування та захисту інформаційних активів, революціонізуючи спосіб, яким ми збираємо та використовуємо інформацію.

Після розгляду ключових аспектів цифровізації та її впливу на сучасні інформаційні системи, ми зосередили увагу на інформаційній розвідці, розкриваючи її значення в контексті обробки та аналізу даних. Ці етапи є фундаментальними для розуміння наступної важливої складової нашого дослідження – OSINT.

OSINT (Open source intelligence) — це метод збору інформації (відкрита розвідка), яка працює із відкритими джерелами для отримання аналітичних, розвідувальних, бізнесових чи інших цілей. Відмінною рисою OSINT є той факт, що інформація збирається із загальнодоступних джерел: соціальних мереж, юридичних документів, ЗМІ, державних записів, фінансових звітів, різноманітних веб-сайтів, професійних журналів, конференцій, блогів.

Ідея OSINT полягає в тому, що велика кількість корисної інформації загальнодоступна та може бути використана для різноманітних цілей. Це особливо важливо у сучасному світі, де інформація має здатність швидко розповсюджуватися через інтернет та соціальні медіа. Наприклад, урядові агентства можуть використовувати OSINT для національної безпеки та контррозвідувальних операцій, використовуючи новинні статті, блоги, соціальні медіа, інших відкриті джерела для виявлення потенційних загроз, розуміння політичних тенденцій різних країн чи відстеження розвитку військових технологій.

Для бізнесу OSINT може бути використана у процесі аналізу конкурентного середовища, ринкових досліджень та виявлення нових можливостей. Наприклад, компанії можуть аналізувати веб-сайти конкурентів, фінансові звіти, патенти та торгові марки, відгуки споживачів та інші загальнодоступні джерела для отримання цінної інформації про конкурентну стратегію, продукти та послуги.

Технічна сторона OSINT включає в себе використання спеціалізованих інструментів та методів для збору, фільтрації та аналізу великих обсягів інформації. Це може включати в себе використання пошукових систем, баз

даних, інструментів для моніторингу соціальних медіа, геолокаційних інструментів та інших цифрових ресурсів.



Рисунок 1.2 – OSINT Frameworks [18]

Ефективне використання OSINT вимагає не тільки здатності знаходити великі обсяги інформації, але й уміння визначати її релевантність та точність. Ключовим аспектом є здатність критично оцінювати джерела, інтерпретувати зібрану інформацію в контексті та інтегрувати її в більшу картину.

Крім того, OSINT включає в себе етичні міркування, оскільки важливо забезпечити законність та етичність процесів збору інформації. Це означає, що інформацію слід збирати та використовувати у спосіб, що відповідає юридичним нормам та стандартам конфіденційності.

OSINT відіграє важливу роль у сучасному інформаційному суспільстві, надаючи індивідам, організаціям та урядам цінні засоби для збору важливої інформації. Це є важливим інструментом в багатьох областях, від розвідки та безпеки до бізнес-аналітики та журналістики. Особливо важливим OSINT стає у світлі швидкого розвитку цифрових технологій та збільшення обсягів доступної онлайн-інформації, що вимагає від фахівців у цій області постійного оновлення знань та навичок.

## **1.2 Взаємодія OSINT з штучним інтелектом**

Розслідування збиття рейсу MH17, яке сталося 17 липня 2014 року над територією України, є одним з найбільш відомих прикладів, де OSINT відіграв ключову роль. Цей літак, що належав Malaysia Airlines, був збитий ракетою під час польоту з Амстердама до Куала-Лумпур, що призвело до трагічної загибелі всіх 298 людей на борту. Ця подія викликала міжнародне засудження та потребувала ретельного та незалежного розслідування.

OSINT зіграв ключову роль у встановленні фактів та обставин катастрофи, особливо у визначенні відповідальних сторін. Одним з головних гравців у цьому процесі став міжнародний колектив журналістів та дослідників Bellingcat. Використовуючи доступні в Інтернеті джерела, такі як соціальні мережі, відео, фотографії, та інші дані, Bellingcat зміг зібрати та проаналізувати ключову інформацію, яка відіграла важливу роль у розслідуванні.

Одним із перших кроків у розслідуванні стало ідентифікування зброї, використаної для збиття літака. Завдяки аналізу фотографій та відеоматеріалів, опублікованих у соціальних мережах, Bellingcat зміг встановити, що ракетна система, яка збила MH17, була російської виробництва - Бук. Далі команда аналізувала велику кількість фотографій та відео, які демонстрували

переміщення цієї системи по території, контрольованій сепаратистами у східній Україні. Вони також використовували зображення з супутників та інші геолокаційні дані для визначення місцезнаходження системи Бук у конкретні часові періоди.



Рисунок 1.3 – Фото установки "Бук", зроблене в Донецьку в день, коли був збитий "Боїнг" [3]

Ключовим елементом розслідування стало використання соціальних медіа для збору свідчень. Bellingcat зміг ідентифікувати профілі в соціальних мережах, пов'язані з російськими військовими та сепаратистами, які брали участь у конфлікті на сході України. Це дозволило їм зібрати цінну інформацію про рух військових частин, їхнє озброєння та інші релевантні деталі.

OSINT також включав аналіз звукових записів, які були доступні в Інтернеті. Це дозволило розпізнати голоси та ідентифікувати окремих осіб, які могли бути причетні до збиття літака. Інший аспект розслідування включав аналіз метеорологічних даних, щоб визначити умови видимості та інші фактори, які могли вплинути на події того дня.





Рисунок 1.4 – Бойовики "днр" фотографуються на тлі уламків збитого літака

[3]

Важливість OSINT у цьому розслідуванні не можна недооцінювати. Він дозволив незалежним дослідникам та журналістам зібрати та аналізувати велику кількість інформації, яка була ключовою для встановлення обставин катастрофи та притягнення винних до відповідальності. Це розслідування стало прикладом того, як сучасні технології та методи можуть бути використані для розкриття істини в умовах, коли традиційні методи розслідування можуть бути обмежені або недоступні.

Взаємодія між OSINT (відкрита джерельна розвідка) та штучним інтелектом (ШІ) відкриває нові перспективи та можливості в сфері збору та аналізу інформації. Особливо важливим це стає в контексті сучасного інформаційного простору, де обсяги даних, доступних у відкритих джерелах, зростають експоненційно. ШІ може значно підвищити ефективність та точність OSINT, надаючи потужні інструменти для обробки, класифікації та аналізу великих даних.

Однією з ключових переваг використання ШІ у OSINT є автоматизація процесів збору даних. ШІ може швидко сканувати, фільтрувати та збирати інформацію з різноманітних відкритих джерел, включаючи соціальні мережі,

форуми, новинні веб-сайти та блоги. Наприклад, за допомогою алгоритмів машинного навчання можна аналізувати тенденції в соціальних мережах, виявляючи ключові слова, фрази та емоційні навантаження, що дає можливість швидко реагувати на зміни громадської думки або відстежувати поширення дезінформації.

Ще один аспект використання ШІ у OSINT – це покращення якості аналізу даних. ШІ може допомогти в ідентифікації взаємозв'язків між різними наборами даних, які можуть бути неочевидними для людського аналітика. Наприклад, аналізуючи дані з різних географічних регіонів та джерел, ШІ може виявити складні патерни, які свідчать про міжнародні тенденції або глобальні загрози. Інтеграція ШІ з OSINT також покращує можливості візуалізації даних. Завдяки алгоритмам машинного навчання та штучного інтелекту можна ефективно візуалізувати великі набори даних, що дозволяє краще розуміти складні інформаційні структури та знаходити інсайти, які можуть бути приховані в сирій інформації.

ШІ також може бути використаний для забезпечення безпеки інформаційних процесів у рамках OSINT. Використовуючи алгоритми штучного інтелекту для моніторингу та аналізу інформаційних потоків, можна ефективно виявляти спроби кібератак або несанкціонованого доступу до даних.

Важливим застосуванням ШІ у OSINT є розробка прогнозних моделей. Наприклад, використовуючи історичні дані та поточні тенденції, ШІ може допомогти прогнозувати майбутні політичні зміни, економічні кризи або соціальні рухи. Це особливо корисно для урядів та міжнародних організацій, які прагнуть адекватно реагувати на глобальні виклики.

Загалом, інтеграція ШІ з OSINT створює потужні можливості для ефективного збору, аналізу та використання відкритих джерел інформації. Вона дозволяє виявляти та аналізувати складні інформаційні структури, що робить цей процес незамінним інструментом у сучасному швидкоплинному інформаційному світі.

Але окрему увагу слід приділити медіа контенту при належній увазі до деталей, він може розповісти значно більше ніж можна собі уявити. Наприклад шеврон, здавалося б, які небезпечні дані він може в собі нести, але для OSINT у поєднанні зі штучним інтелектом, будь-які дані можуть бути критично важливі. Наприклад до мережі потрапили декілька фотографій на яких знаходяться військові з однаковими шевронами, але окрім цього ніяких критичних даних визначено не було, проте система чи людина зробила помітку по ним. Через деякий час, ЗМІ або наприклад волонтери викладають фотографію, яку підписують "Вчора була відправлена допомога на Запорізький напрямок" і це додає можливість вважати що попередні фотографії були зроблені вже на конкретній місцевості. А ті об'єкти на фото чи навіть рослинність, які до цього моменту не додавалась до аналізу, через обширну область для пошуку, можуть призвести до більш точного визначення місця розташування. Також вже є алгоритми, які за тінню від сонця, можуть визначати пору року та навіть приблизний час, коли була зроблена фотографія.

Отже, якщо розглядати окрему фотографію, вона може не дати розвіднику взагалі ніякої інформації, але при пошуку зв'язків з іншим медіаконтентом чи інформацією може розповісти дуже багато.

### **1.3 Постановка задачі для реалізації**

Навчання моделі комп'ютерного зору, виявляти повністю усі вразливі об'єкти на фото (наприклад техніки, зброї, частини будівель), потребує залучення великої команди спеціалістів, оскільки потребує набагато детального аналізу об'єктів, збору та обробки зображень, їх анотацій та значних потужностей для створення моделі штучного інтелекту. Тому в рамках роботи, я обмежуватиме виявленням шевронів та нашивок з групною крові і позивним. А також їх зафарбовуванням або розмиттям, з метою захисту від потенційного витоку. Завдання проекту включають наступне:

- 1) Навчання моделі штучного інтелекту для автоматичного пошуку шевронів та нашивок з групною крові і позивним на фото. Забезпечення якомога вищої точності виявлення.
- 2) Розробка веб додатку та інтеграція моделі штучного інтелекту до нього.
- 3) Перевірка можливості обробки даних на стороні клієнта, без відправки фотографії користувача на сервер. Для забезпечення конфіденційності та безпеки інформації.
- 4) Розробка інтерфейсу користувача. Створення зручного та інтуїтивно зрозумілого інтерфейсу, що дозволить користувачам завантажувати та отримувати оброблені зображення.

## 2 НАВЧАННЯ МОДЕЛІ

### 2.1 Огляд алгоритмів комп'ютерного зору для виявлення об'єктів

Проаналізувавши різні моделі комп'ютерного зору глибокого навчання (Deep Learning), я визначив, що алгоритми виявлення об'єктів поділяються на дві категорії (рис 2.1) залежно від того, скільки разів одне й те саме вхідне зображення пропускається через мережу. Отже існують одноступеневі та двоступеневі детектори. Одноступеневі детектори є більш простими та ефективними, ніж двоступеневі детектори. Вони використовують один етап для виявлення та класифікації об'єктів, що робить їх швидшими та менш витратними, однак вони також є і менш точними. Але оскільки обробка зображення в проєкті буде відбуватися на стороні клієнту, в першу чергу цікавить використання ресурсів, отже двоступеневі розглядатись не будуть.

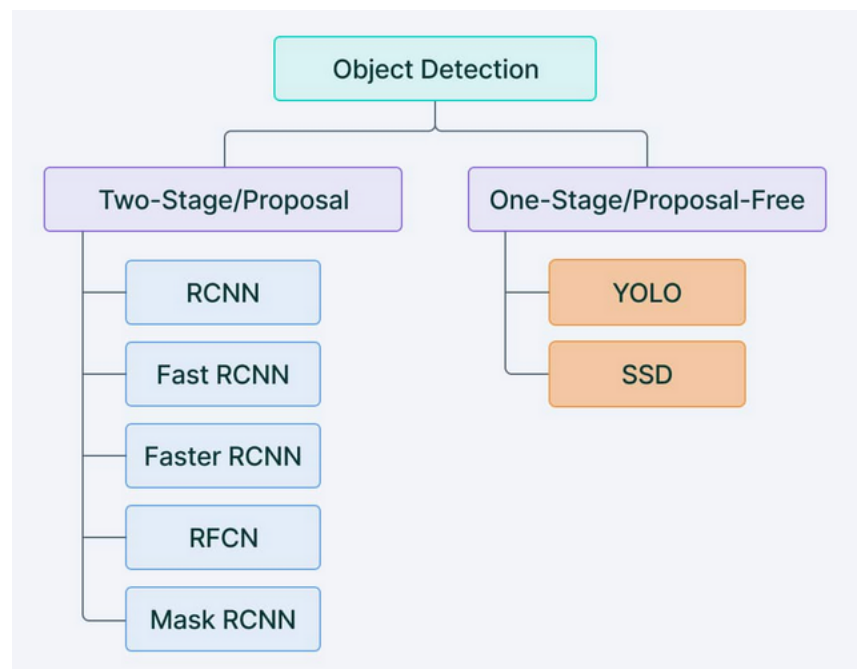


Рисунок 2.1 – Одно- та двоступеневі алгоритми [21]

**YOLO** (You Only Look Once) – ця потужна модель штучного інтелекту, яка використовується для виявлення об'єктів на зображеннях або відео (навіть в реальному часі). YOLO була представлена у 2016 році Джозефом Редмоном та

ін., вона здійснила революцію у виявленні об'єктів завдяки своїй швидкості та точності. YOLO застосовує гарний підхід до виявлення об'єктів, використовуючи глибоку згорткову нейронну мережу (CNN). Продуктивність моделі оцінюється за допомогою таких показників, як перетин об'єднання (IoU) та середня точність (AP). IoU вимірює точність локалізації, тоді як AP оцінює, наскільки добре модель розпізнає та класифікує об'єкти. Що до механізму роботи, то архітектура YOLO розбиває зображення на решітку, де кожна сітка відповідає за виявлення та локалізацію об'єктів у своєму секторі, прогножуючи координати обмежувальної рамки та інформацію про об'єкт. Це зменшує обчислювальні потреби, оскільки і виявлення, і розпізнавання виконуються одночасно. Швидкість обчислень та простота YOLO зробили його придатним для додатків у режимі реального часу, таких як відеоспостереження та автомобілі з автопілотом. Порівняння п'ятої версії YOLO, з останньою восьмою, можна побачити на (рис 2.2).

## Object Detection Performance Comparison (YOLOv8 vs YOLOv5)

Model Size	YOLOv5	YOLOv8	Difference
Nano	28	37.3	+33.21%
Small	37.4	44.9	+20.05%
Medium	45.4	50.2	+10.57%
Large	49	52.9	+7.96%
Xtra Large	50.7	53.9	+6.31%

\*Image Size = 640

Рисунок 2.2 – Порівняння моделей YOLO версій 5 та 8 [22]

**SSD** (Single Shot MultiBox Detector) була розроблена у 2016 році Вей Лю та іншими дослідниками. Основна ідея SSD полягає в тому, щоб виконувати виявлення об'єктів та їх класифікацію в одному проході, звідси й назва "Single Shot". SSD також використовує глибокі згорткові нейронні мережі (CNN), але на відміну від YOLO вона працює з декількома згортковими шарами, результати яких інтерпретуються як обмежувальні рамки та класи об'єктів. Через це SSD працює повільніше, але більш точноше ніж YOLO.

Таблиця 2.1 – Порівняння YOLO та SSD

<b>Властивість</b>	<b>YOLO</b>	<b>SSD</b>
Архітектура	Одноступенева	Одноступенева
Рівень абстракції	Піксельний	Піксельний
Точність	Середня	Висока
Швидкість	Швидка	Середня
Складність реалізації	Проста	Середня
Обчислювальна складність	Низька	Середня

Оскільки планується реалізація на клієнтській частині, а точність та швидкодія розпізнавання у обох моделей на рівні, то вирішальним параметром буде обчислювальна складність, тобто залежність від потужності обчислювальної техніки. Отже, для навчання моделі, будемо використовувати YOLO.

Розглянемо детальніше, як він працює та його архітектуру. Алгоритм YOLO отримує зображення на вході, а потім використовує просту нейронну мережу глибокої згортки для виявлення об'єктів на зображенні. Архітектура моделі CNN, яка є основою YOLO, показана на (рис.2.3).

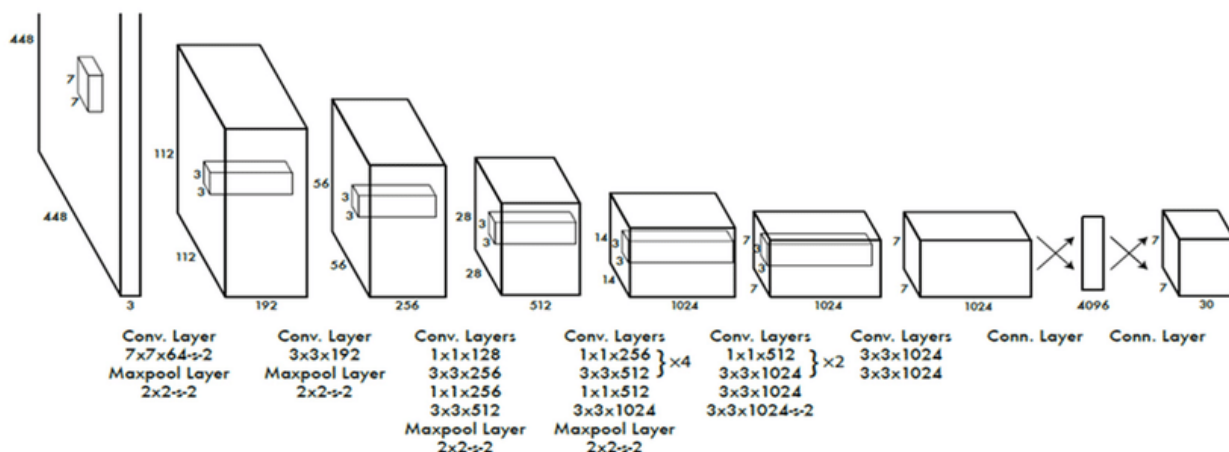


Рисунок 2.3 – Зображення CNN архітектури [23]

Перші 20 шарів згортки моделі попередньо навчаються за допомогою ImageNet, шляхом додавання тимчасового середнього об'єднання і повністю зв'язаного шару. Потім ця попередньо навчена модель конвертується для виконання виявлення, оскільки попередні дослідження показали, що додавання згортки та зв'язаних шарів до попередньо навченої мережі покращує продуктивність. Останній повністю зв'язний шар YOLO прогнозує як ймовірності класів, так і координати обмежувальних рамок.

YOLO ділить вхідне зображення на сітку  $S \times S$ . Якщо центр об'єкта потрапляє в комірку сітки, ця комірка відповідає за виявлення цього об'єкта. Кожна комірка сітки прогнозує змінну кількість обмежувальних рамок і довірчі оцінки для цих рамок. Ці оцінки впевненості відображають, наскільки модель впевнена, що область містить об'єкт, і наскільки точною, на її думку, є передбачена область.

YOLO прогнозує декілька обмежувальних рамок на кожну клітинку сітки. Під час навчання ми хочемо, щоб за кожен об'єкт відповідав лише один предиктор граничної області. YOLO призначає один предиктор "відповідальним" за прогнозування об'єкта, на основі якого прогноз має найвищу поточну IOU з базовою істиною. Це призводить до спеціалізації між предикторами граничної області. Кожен предиктор стає кращим у прогнозуванні



певних розмірів, співвідношень сторін або класів об'єктів, покращуючи загальну оцінку пригадування.

Однією з ключових технік, що використовуються в моделях YOLO, є не-максимальне придушення (NMS). NMS - це етап постобробки, який використовується для підвищення точності та ефективності виявлення об'єктів. При виявленні об'єктів для одного об'єкта на зображенні зазвичай генерується кілька обмежувальних рамок. Ці рамки можуть перекриватися або розташовуватися в різних місцях, але всі вони представляють один і той самий об'єкт. NMS використовується для виявлення та видалення надлишкових або неправильних рамок, а також для виведення однієї рамки для кожного об'єкта на зображенні.

## **2.2 Пошук, підготовка та анотація зображень**

Перший кроком у навчанні своєї моделі є анотація об'єктів на фотографіях, в моєму випадку, це розмітка шевронів та нашивок з групною крові і позивним. Зображення я збирав з відкритих джерел, таких як різні групи у соціальних мережах, зокрема у Facebook, публічних чатах та групах різних волонтерів.

Критерії вибору зображень були такими:

- різноманітність. Зображення повинні відображати різноманітність сцен, об'єктів, освітлення, кольорів і перспектив, щоб модель могла ефективно вчитися і бути стійкою до різних умов;
- якість та роздільна здатність. Чим краща буде якість зображення, тим краще навчитися модель. Але також потрібно мати і фото з деякими відхиленнями (розмиті або з шумом), щоб модель могла впоратись з недосконалими умовами;
- баланс між об'єктами. Для того, щоб уникнути упередженості моделі до об'єктів які більш всього зустрічаються;
- зображення які не містять об'єкти які потрібно виявляти. Наявність зображень без цільових об'єктів допомагає моделі вчитися правильно

ідентифікувати випадки, коли об'єкт відсутній. Це зменшує ймовірність помилкових позитивів (коли модель помилково визначає об'єкт там, де його немає).

Для анотацій я використовував сервіс [roboflow.com](https://roboflow.com), він дозволяє зробити це дуже зручно і швидко. Також серед його переваг, є те що він дозволяє зробити багатокутні анотації (рис. 2.4).

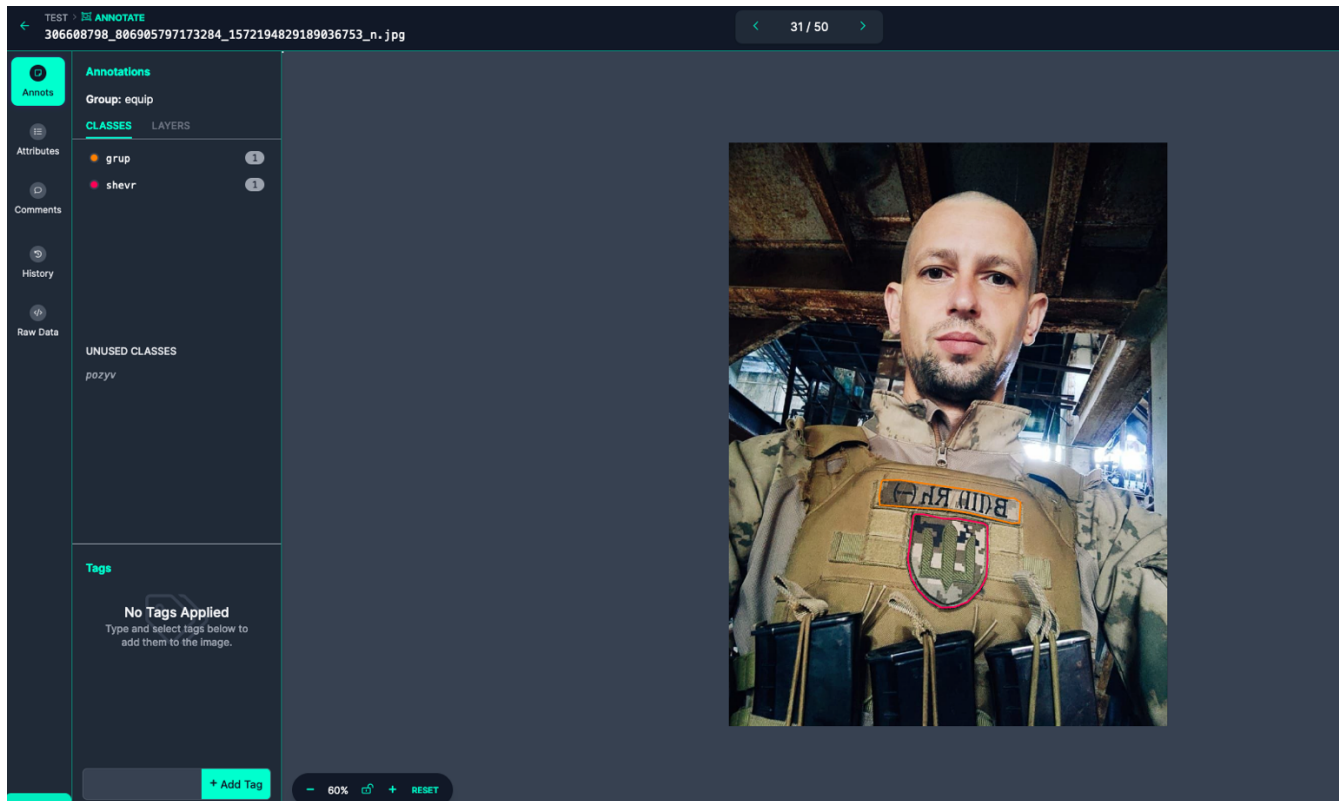


Рисунок 2.4 - Багатокутні анотації в Roboflow

Достатньо натиснути на зображення і вказівником миші пройтись по рамках об'єкту, фіксуючи її в потрібному місці кліком, а після обрати клас, тобто який саме об'єкт ми виділили. Вони мають вагоме значення для підготовки наборів даних для навчання моделі та часто можуть підвищити її точність, якщо використовуються замість звичайних прямокутних рамок, але без належних інструментів (в даному випадку сервіс Roboflow) для нанесення міток потрібно більше часу.

Ще однією з його зручних можливостей, є імпорт створеного dataset в один з популярних форматів (рис 2.5) в нашому випадку одразу у формат YOLO. Це означає, що не потрібно витратити час на ручне перетворення форматів або налаштування складних конфігурацій. Ця функція робить Roboflow ідеальним інструментом для дослідників, інженерів та розробників, які працюють над проектами об'єктного виявлення, дозволяючи їм зосередитися більше на розробці та вдосконаленні своїх моделей, а не на технічних аспектах підготовки даних.

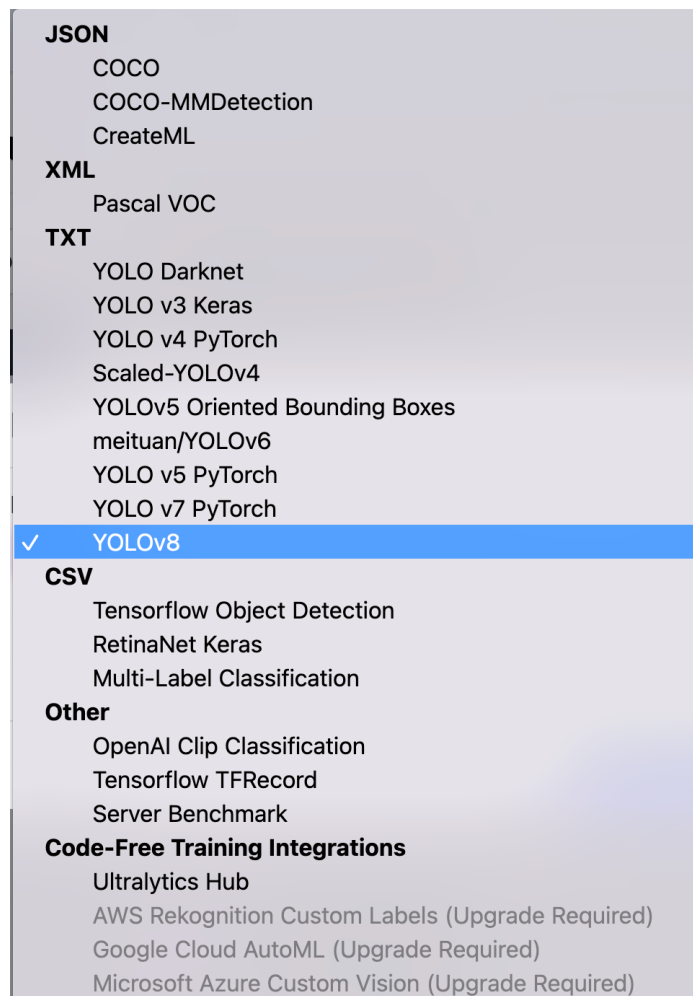


Рисунок 2.5 – Формати імпорту dataset у сервісі Roboflow

Після завершення розмітки анотацій, необхідно розділити зображення на тренувальну, валідаційну та тестову вибірки. Для тренувального сету я виділив 71% зображень, для валідаційного та тестового по 14%. Після цього привести зображення до одного розміру, в моєму випадку я обрав 640x640 пікселів.

Завершальною для датасету дією буде аугментація - техніка, яка створює додаткові тренувальні дані з існуючого набору даних шляхом внесення випадкових змін або трансформацій (рис 2.6), наприклад повороти, масштабування, відбиття або зміни освітлення зображень, .

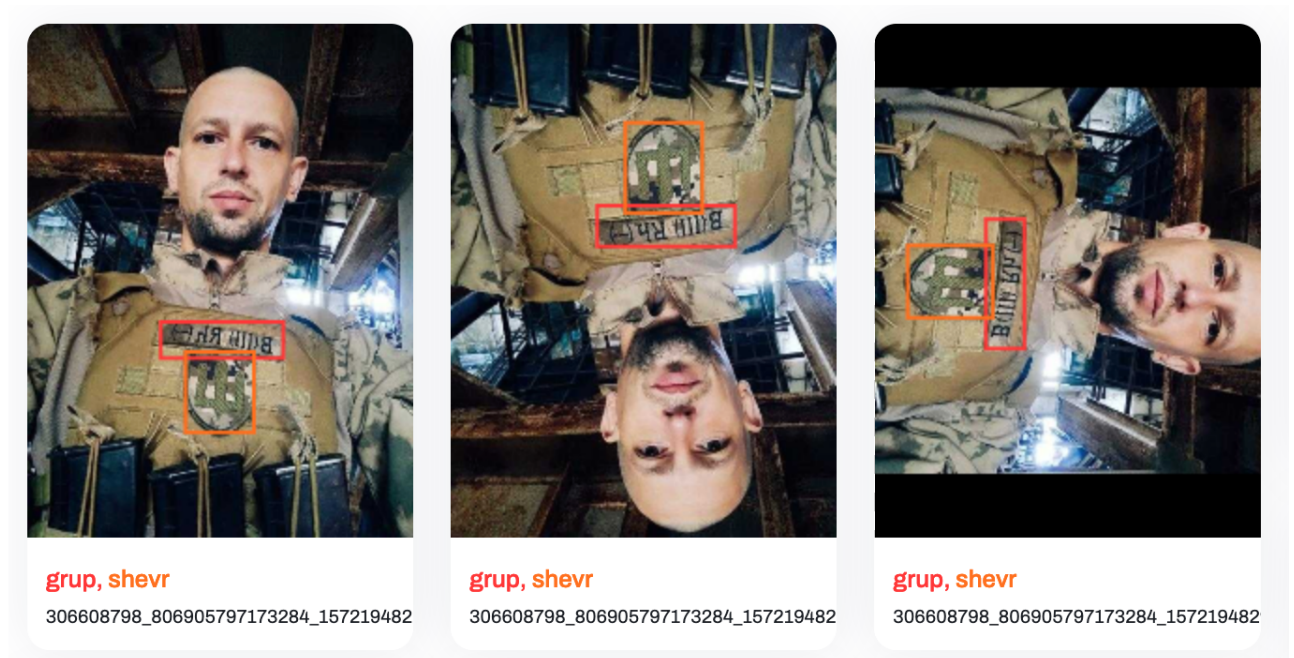


Рисунок 2.6 – Приклад аугментації

Аугментація даних допомагає моделям штучного інтелекту стати більш точними, за рахунок навчання розпізнавати об'єкти в різних умовах, зменшує ризик перенавчання на обмеженому наборі даних та покращує точність та надійність моделей. Типи аугментацій, які я задіяв:

- Перевертання (Flip):
  1. Горизонтальне перевертання: Кожне зображення в наборі даних буде відображене по горизонталі, створюючи дзеркальне відображення зображення.
  2. Вертикальне перевертання: Зображення буде перевернуте по вертикалі, створюючи дзеркальне відображення зверху вниз.
- Поворот на 90° (90° Rotate):

1. Поворот за годинниковою стрілкою: Зображення повертається на 90 градусів за годинниковою стрілкою.
2. Поворот проти годинникової стрілки: Зображення повертається на 90 градусів проти годинникової стрілки.

- Обрізання (Crop):

1. Мінімальне збільшення 0%: Немає обрізання, зображення залишається в початковому розмірі.
2. Максимальне збільшення 20%: Зображення обрізається так, що воно збільшується на 20%, фокусуючись на центральній частині зображення.

Отже, вихідний датасет має наступні дані (таблиця 2.2).

Таблиця 2.2 - Дані отриманого датасету

Кількість знайдених мною зображень	110
Кількість зображень після аугментації	210
Розподілення Train / Valid / Test	71% / 14% / 14%
Розмір зображень	640 x 640 пікселів

### 2.3 Навчання моделі

Серед моделей YOLOv8 для операції розпізнавання доступні наступні моделі (рис. 2.7).

Model	size (pixels)	mAP <sup>val</sup> <sub>50-95</sub>	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Рисунок 2.7 – Моделі YOLOv8 для розпізнавання [24]

В таблиці 2.7 Model вказує на назву моделі, а остання буква на її розмір. Так "n" вказує на розмір nano, "s" – small, "m" – medium, "l" – large, та "x" – extra large. Size відповідає за розмір зображень на яких навчали моделі, в даному випадку у всіх моделей значення 640 пікселів, що вказує на роздільну здатність 640x640. mAP (mean average precision) – це показник, який використовується для оцінки виявлення об'єктів. Тобто він є мірою точності моделі та вказує на середній відсоток випадків, коли модель правильно розпізнавала об'єкти на зображенні. Його формула базується на наступних під показниках:

- Confusion Matrix;
- Intersection over Union (IoU);
- Recall;
- Precision.

Speed CPU ONNX – швидкість обробки моделі на процесорі за допомогою бібліотеки ONNX Runtime. У стовпці Speed A100 TensorRT вказана швидкість обробки моделі на графічному процесорі Nvidia A100 за допомогою бібліотеки TensorRT. Можна наглядно побачити, як домінує швидкість обчислень на графічному чипі. Params відповідає за кількість параметрів моделі в мільйонах. Більша кількість параметрів може вказувати на більшу здатність до навчання, але також може збільшувати вимоги до обчислювальних ресурсів. FLOPs - кількість операцій з плаваючою комою в секунду в мільярдах, необхідних для одного проходження вперед (forward pass) моделі. Це показник обчислювальної складності моделі. Тож доцільним вибором моделі, яка працюватиме у web-додатку через браузер буде "найлегша" модель – nano, але також перевіримо і small модель і порівняємо їх швидкодію.

Для навчання основної моделі будуть використані попередньо навчені моделі - остання і найактуальніша на цей час 8-ма версія YOLO та декілька моделей 5-ої версії, які рекомендує компанія Ultralytics. Нижче наведена порівняльна таблиця 2.3 цих моделей.

Таблиця 2.3 – Порівняння обраних моделей

<b>Модель</b>	<b>Точність</b>	<b>Швидкість</b>
YOLOv8n	37.3%	80.4 мс
YOLOv5u	34.3%	73.6 мс
YOLOv5u6	42.1%	211 мс

Для проведення навчання моделей було використано сервіс Google Colaboratory який є безкоштовною хмарною службою, яка надається Google для сприяння науковим дослідженням та освіті. Основна його ідея полягає в тому, щоб зробити машинне навчання доступнішим для широкої публіки. Ця платформа дозволяє користувачам писати та виконувати код Python безпосередньо в браузері, що великою мірою спрощує процес, адже не потрібно турбуватися про налаштування та управління робочим середовищем. Google Colab надає безкоштовний доступ до обчислювальних ресурсів, включно з графічними процесорами (GPU) та тензорними процесорами (TPU), що є важливим для завдань, пов'язаних з глибоким навчанням. Ще одна ключова перевага Colab полягає у спільній роботі. Користувачі можуть легко ділитися своїми блокнотами Jupyter, які використовуються в Colab, з іншими особами, що сприяє співпраці та спільному розв'язанню задач. Це робить Colab ідеальним інструментом для освітніх проектів, спільних досліджень та розробки в галузі машинного навчання.

Використання Colab також включає інтеграцію з іншими сервісами Google, такими як Google Drive, що забезпечує зручне зберігання та доступ до даних. Він підтримує бібліотеки Python, які є стандартом у наукових та аналітичних дослідженнях, що робить його корисним інструментом для аналізу даних, статистики, візуалізації, та інших областей.

Почнемо навчати обрані вище моделі, у розмірі “nano”. Після навчання моделі YOLOv8n, отримали наступні результати (таблиця 2.4).

Таблиця 2.4 – Навчання моделі YOLOv8n

Class	Images	Instances	P	R	mAP50	mAP50-95
all	22	59	0.359	0.192	0.263	0.149
grup	22	6	0	0	0.0163	0.0061
pozyv	22	25	0.483	0.263	0.38	0.192
shevr	22	28	0.593	0.312	0.393	0.249

Загалом модель показала непогані результати, але є деякі недоліки. Так, для класу "Група" mAP50 і mAP50-95 дорівнюють 0, що означає, що модель не змогла виявити жодного примірника цього класу. Для класу "Позив" mAP50 також низький, що означає, що модель не завжди може точно виявити цей клас. Найкращі результати модель показала для класу "Шевр", де mAP50 і mAP50-95 дорівнюють 0,393 і 0,249 відповідно.

Перейдемо до навчання другої моделі YOLOv5u (таблиця 2.5).

Таблиця 2.5 – Навчання моделі YOLOv5u

Class	Images	Instances	P	R	mAP50	mAP50-95
all	22	59	0.35	0.29	0.318	0.182
grup	22	6	0	0	0.0431	0.0214
pozyv	22	25	0.36	0.44	0.422	0.237
shevr	22	28	0.689	0.429	0.49	0.287

Загалом модель показала кращі результати, ніж у попередньому випадку. Так, для класу "Група" mAP50 дорівнює 0,0431, що означає, що модель все ж таки змогла виявити один примірник цього класу. Для класу "Позив" mAP50 також трохи зріс, що означає, що модель стала краще виявляти цей клас. Найкращі результати модель показала для класу "Шевр", де mAP50 і mAP50-95 дорівнюють 0,49 і 0,287 відповідно.



Перейдемо до навчання третьої моделі YOLOv5u6 (таблиця 2.6).

Таблиця 2.6 – Навчання моделі YOLOv5u6

Class	Images	Instances	P	R	mAP50	mAP50-95
all	22	59	0.841	0.0373	0.117	0.0851
grup	22	6	1	0	0.00156	0.000514
pozyv	22	25	0.523	0.0455	0.13	0.0821
shevr	22	28	1	0.0665	0.219	0.173

Ця модель показала найгірші результати, майже в три рази гірше ніж попередня модель.

Тепер розглянемо графік, для перевірки, чи не була перенавчена модель. Перенавчання, це звикання моделі до тренувального датасету, коли вона занадто добре «вивчила» тренувальний набір даних, до такої міри, що вона запам'ятовує конкретні дані та їхні шуми, замість того, щоб узагальнити правила, які можна було б застосувати до нових даних. Для усіх результатів проведених до цього, кількість епох дорівнювала 100, перенавчання судячи з графіку на рис. 2.8 не було. Ознаками перенавчання є раптове падіння до дуже низького рівня або початок збільшення значень. Ось деякі спостереження щодо Box Loss. Значення втрат на тренувальному наборі (синя лінія) зменшуються з часом та стабілізуються, що є хорошим знаком. Втрати на валідаційному наборі (помаранчева лінія) також зменшуються, але їхня тенденція більш волатильна, що може свідчити про деяку невизначеність у валідаційних даних. Щодо Class Loss, то цей графік показує хорошу конвергенцію між тренувальними та валідаційними втратами. Значення втрат для тренувального набору знижуються та стають дуже близькими до втрат валідаційного набору, що є індикатором того, що перенавчання не відбувається.

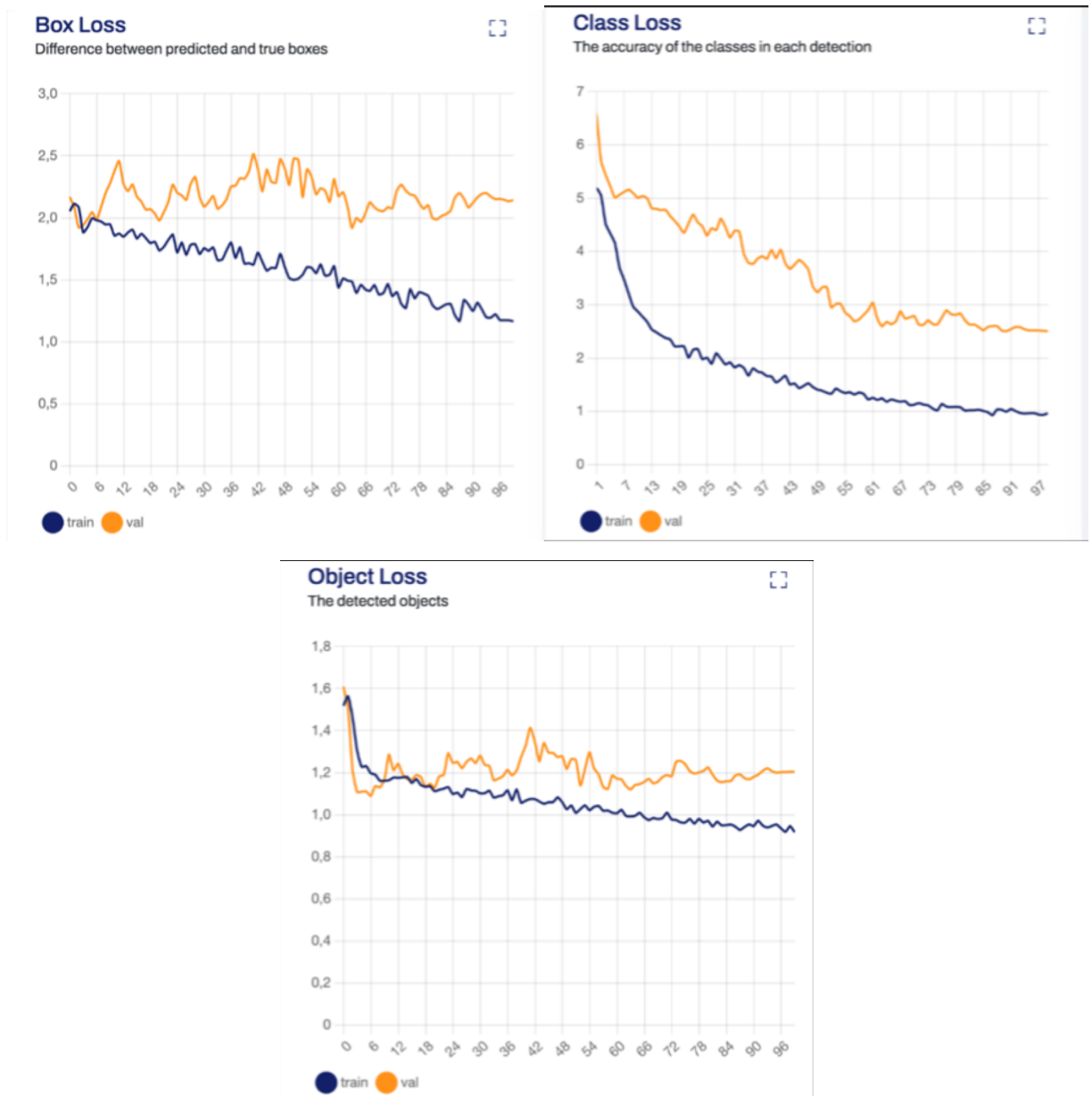


Рисунок 2.8 – Графіки точності сформовані в процесі навчання

Object Loss, показує себе аналогічно до "Class Loss", втрати на тренувальному наборі зменшуються і збігаються з втратами на валідаційному наборі. Немає значного розриву між двома наборами, що також є позитивним знаком. Отже збільшимо кількість епох. Збільшивши кількість епох до 150, помічаємо на графіку (рис. 2.9), що приблизно з 135 епохи відбувається зростання значень, тож оптимальним значенням епох для нашого датасету буде до 135.



Рисунок 2.9 – Навчання моделі на 150 епохах

Перенавчимо модель YOLOv5u, на 130 епохах (таблиця 2.7).

Таблиця 2.7 – Модель YOLOv5u на 130 епохах

Class	Images	Instances	P	R	mAP50	mAP50-95
all	22	59	0.352	0.31	0.322	0.19
grup	22	6	0	0	0.034	0.0126
pozyv	22	25	0.421	0.495	0.448	0.26
shevr	22	28	0.635	0.435	0.482	0.297

Точність покращилась, але не на багато, наприклад параметр mAP50 був 0.318, а став 0.322; параметр mAP50-95 був 0.182, став 0.19. Ці покращення не значні, але обов'язким тут було впевнитись, що модель не перенавчається.

## 3 РОЗРОБКА ДОДАТКУ

### 3.1 Вибір фреймворків

#### Фреймворк для моделі штучного інтелекту

Для запуску моделі на клієнтській частині, я обрав TensorFlow.js. TensorFlow.js є надзвичайно потужним інструментом для розробників, які прагнуть використовувати можливості машинного навчання у веб-додатках. Цей фреймворк відкриває нові горизонти для інтеграції штучного інтелекту у веб-простір, дозволяючи реалізовувати інноваційні рішення безпосередньо в браузері користувача. Однією з ключових переваг TensorFlow.js є його універсальність і гнучкість. Розробники можуть не тільки використовувати вже натреновані моделі машинного навчання, але й тренувати нові моделі прямо у браузері. Це відкриває широкі можливості для створення інтерактивних додатків, які адаптуються та вчаться на основі даних користувача в реальному часі.

Завдяки інтеграції з JavaScript, TensorFlow.js є зручним та доступним вибором для широкого кола розробників. JavaScript є однією з найпопулярніших мов програмування, і багато веб-розробників вже знайомі з нею. TensorFlow.js легко вписується в існуючі веб-проекти, дозволяючи розробникам без зайвих зусиль додавати функції машинного навчання до своїх додатків.

Важливою перевагою TensorFlow.js є його спільнота та підтримка. Велика та активна спільнота розробників не тільки сприяє швидкому вирішенню проблем, але й постійно розробляє нові інструменти та бібліотеки, які роблять роботу з TensorFlow.js ще більш продуктивною та цікавою.

Одним з ключових аспектів TensorFlow.js є забезпечення приватності даних. Оскільки обробка даних відбувається локально на пристрої користувача, це мінімізує ризики пов'язані з передачею даних на зовнішні сервери. Це особливо важливо у контексті зростаючих вимог до захисту персональних даних.

Ефективність є ще однією важливою перевагою TensorFlow.js. Фреймворк оптимізований для використання сучасних веб-технологій, таких як WebGL, що

дозволяє ефективно виконувати складні обчислення на пристроях користувачів. Це означає, що навіть складні моделі машинного навчання можуть працювати плавно та ефективно в браузері.

Загалом, TensorFlow.js є надзвичайно цінним інструментом для розробників, які прагнуть використовувати передові технології машинного навчання у своїх веб-додатках. Його гнучкість, легкість інтеграції, активна спільнота та фокус на приватності даних роблять його ідеальним вибором для розробки інноваційних веб-рішень.

### **Фреймворк для інтерфейсу користувача**

Для створення інтерфейсу користувача я обрав Vue.js, оскільки він є одним з провідних фреймворків для розробки веб-інтерфейсів, який завоював популярність серед розробників завдяки своїй гнучкості, легкості вивчення та ефективності. На перше місце у перевагах Vue.js виходить його простота та зручність у використанні. Цей фреймворк пропонує чистий та зрозумілий синтаксис, що робить процес навчання набагато легшим, особливо для новачків у сфері фронтенд-розробки. Vue.js дозволяє швидко створювати інтерактивні веб-інтерфейси, використовуючи декларативний підхід до рендерингу даних у DOM. Ще однією важливою перевагою Vue.js є його реактивність. Система реактивності фреймворка забезпечує автоматичне оновлення інтерфейсу користувача при зміні даних, що робить розробку динамічних веб-додатків більш інтуїтивною та ефективною. Vue.js також відомий своєю гнучкістю та модульністю. Розробники можуть легко інтегрувати Vue.js з іншими бібліотеками або існуючими проектами, а також використовувати його для створення складних односторінкових додатків (SPA). Ця гнучкість робить Vue.js підходящим для широкого спектра проектів, від невеликих веб-сайтів до великих масштабних веб-додатків. Компонентний підхід Vue.js є ще однією його сильною стороною. Розробка з використанням компонентів дозволяє розділити інтерфейс користувача на незалежні, перевикористовувані блоки, що сприяє більш організованому та легшому для підтримки коду. Vue.js також підтримує

широкий спектр сучасних інструментів та екосистему, що включає такі інструменти, як Vue Router для маршрутизації та Vuex для управління станом. Це забезпечує розробникам потужні можливості для створення професійних веб-додатків. Важливою перевагою Vue.js є його активна та дружня спільнота, що надає велику кількість навчальних матеріалів, ресурсів та підтримки. Це робить процес навчання та розробки з Vue.js більш приємним та доступним.

Підсумовуючи, Vue.js є відмінним вибором для розробників, які шукають ефективний, легкий у вивченні та гнучкий фреймворк для створення сучасних веб-інтерфейсів. Його простота, реактивність, модульність та сильна спільнота роблять його ідеальним інструментом для різноманітних веб-проектів.

### 3.2 Опис програмної реалізації

Додаток складається з файлів зображених на рис. 3.1. Файлом який відповідає за ініціалізацію та монтування кореневого компонента Vue до DOM є main.js.

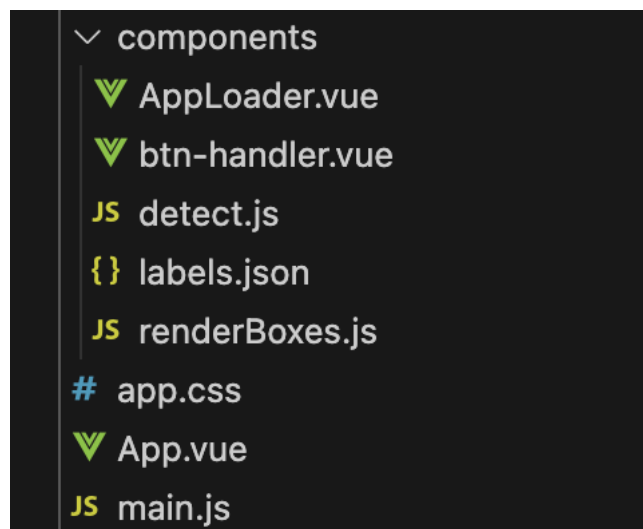


Рисунок 3.1 – Файли додатку

Основним компонентом Vue є файл App.vue. Він служить як основний контейнер для всіх інших елементів. Розберемо функції та бібліотеки, що були імпортовані (рис 3.2).

```
import { toRaw } from 'vue';
import * as tf from '@tensorflow/tfjs';
import '@tensorflow/tfjs-backend-webgl';
import AppLoader from './components/AppLoader';
import ButtonHandler from './components/btn-handler';
import { detect } from './components/detect';
import './app.css';
```

Рисунок 3.2 – Імпорт модулів

Функція `toRaw` використовується для перетворення реактивних об'єктів `Vue` назад в їх вихідні не-реактивні стани. `Tensorflow/tfjs` є бібліотекою машинного навчання для JavaScript. `Tfjs-backend-webgl` імпортує `WebGL backend`, який дозволяє `TensorFlow.js` виконувати операції на графічному процесорі для прискорення обчислень. `AppLoader` відповідає за анімацію стартового завантаження додатку. `ButtonHandler` використовується для обробки взаємодії з кнопкою завантаження зображення. `Detect.js` містить ключову логіку для обробки зображень з використанням `TensorFlow` у додатку. Цей файл включає функції для попередньої обробки зображень, виявлення об'єктів та відображення результатів на канвасі. `App.css` відповідає за стилі відображення.

Розглянемо детальніше файл `detect.js`. Функція `preprocess` (рис. 3.3) відповідає за попередню обробку зображення. Функція отримує на вхід параметри `source` – це вхідне зображення, `modelWidth` та `modelHeight` – ширину та висоту моделі, до якої буде масштабоване зображення. Потім зображення конвертується у тензор - багатовимірний масив, який `TensorFlow.js` використовує для обробки даних, визначаються висота та ширина зображення, обчислюється максимальний розмір `maxSize`, далі зображення доповнюється `img.pad` до квадратного формату, додаючи пікселі знизу або справа відповідно до різниці між `maxSize` та вимірами зображення.

```

const preprocess = (source, modelWidth, modelHeight) => {
  let RatioX, RatioY; // співвідношення

  const input = tf.tidy(() => {
    const img = tf.browser.fromPixels(source);

    // вирівнювання зображення до квадрата => [n, m] to [n, n], n > m
    const [h, w] = img.shape.slice(0, 2); // отримати ширину та висоту джерела
    const maxSize = Math.max(w, h); // отримати максимальний розмір
    const imgPadded = img.pad([
      [0, maxSize - h], // тільки знизу
      [0, maxSize - w], // тільки справа
      [0, 0],
    ]);

    RatioX = maxSize / w; // оновлюємо RatioX
    RatioY = maxSize / h; // оновлюємо RatioY
    console.log(RatioX, RatioY)

    return tf.image
      .resizeBilinear(imgPadded, [modelWidth, modelHeight]) // змінити розмір фрейму
      .div(255.0) // нормалізація
      .expandDims(0);
  });

  return [input, RatioX, RatioY];
};

```

Рисунок 3.3 – Функція preprocess файлу detect.js

Після цього обчислюються співвідношення RatioX і RatioY, які визначають наскільки відрізняються виміри оригінального зображення від maxSize, це необхідно для коректного відображення результатів виявлення на оригінальному зображенні. Функція resizeBilinear змінює розмір зображення до вказаних розмірів моделі, а далі метод div нормалізує зображення, перетворюючи значення пікселів із діапазону [0, 255] у діапазон [0, 1]. expandDims додає додатковий вимір до тензора, який потрібен для подання даних у вигляді пакету (batch) даних. Повертає ж функція – оброблене зображення input та знайдене співвідношення сторін, які будуть використані потім для масштабування та розташування результатів виявлення на оригінальному зображенні.

Тепер перейдемо до основної функції detect, яка є ключовою для процесу виявлення об'єктів на зображеннях. Вона використовує модель TensorFlow.js для



аналізу зображень та відображає результати на канвасі. На вхід функція отримує source вхідне зображення для аналізу, модель що буде використовуватись для виявлення об'єктів, canvasRef – посилання на елемент канваса, на якому будуть відображатися результати та функцію зворотного виклику callback яка буде виконуватись після завершення виявлення. Вже в тілі функції tf.engine().startScope ініціює область виділення пам'яті, а далі викликається функція preprocess, яка описана вище. Після цього модель застосовується до обробленого зображення і результат res транспонується для подальшої обробки. Далі вираховуються границі об'єктів (x1, y1, x2, y2), класифікуються та оцінюється достовірність виявлення. Потім використовується фільтрація з використанням Non-Maximum Suppression (описано у другій частині) для відбирання "найкращих" границь та уникнення повторів та накладань. Дані про границі boxes\_data, оцінки scores\_data та класи classes\_data відібраних об'єктів витягаються для подальшого відображення на канвасі з використанням renderBoxes. Після обробки та відображення результатів, пам'ять, використана для тимчасових тензорів, очищується, а область виділення пам'яті TensorFlow.js закривається за допомогою функції tf.engine().endScope

### 3.3 Аналіз результатів

Перевіримо роботу додатку та навченої моделі зображенням, якого не було в датасеті. Важливо сказати, що зараз в коді стоїть налаштування для порогу оцінку – 20%, тобто якщо модель комп'ютерного зору впевнена на 20%, що вона знайшла об'єкт вірно – вона маскує його. Зі збільшенням розміру датасету, це значення можна буде збільшити. Обираємо зображення натискаючи на кнопку «Open Image»(рис 3.4).

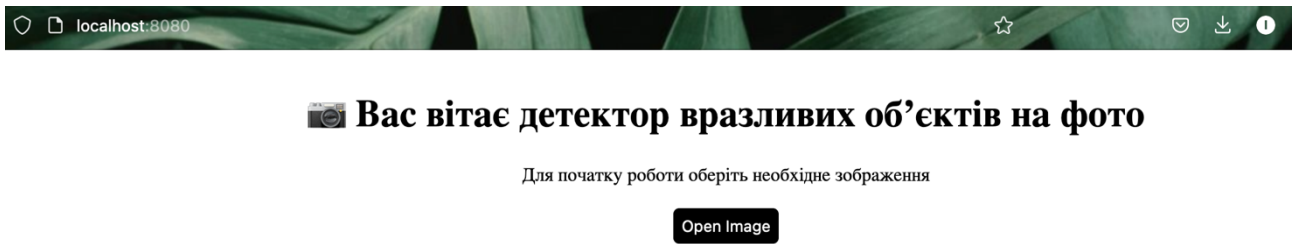


Рисунок 3.4 – Вибір зображення

Розглянемо фото, яке я збираюсь завантажити у додаток, на ньому є два шеврони на лівій руці та нашивка з групою крові на спині (рис 3.5).



Рисунок 3.5 – Зображення з військовим

Бачимо (рис 3.6), що система впоралась на відмінно, замалювавши потрібні об'єкти.

## Вас вітає детектор вразливих об'єктів на фото

Для початку роботи оберіть необхідне зображення



Close Image

Рисунок 3.6 – Результат роботи системи

Візьмемо складніше зображення, а також нижчої якості (рис 3.7).



Рисунок 3.7 – Інше зображення для тестування системи

Система також добре спрацювала, але не змогла знайти шеврон розташований на лівій руці військового. Можливо через те, що воно повернуте під гострим кутом або можливо в датасеті подібних зображень для навчання не було, або їх виявилось не достатньо.

Підводячи підсумок, можна сказати, що відчувається недостатня кількість зображень в датасеті, але можливості YOLO вражають, зі 110 зображень (не додаючи сюди аугументацію) ми отримали гарно працюючу модель. Зрозуміло, що для повноцінної роботи системи в реальних умовах кількість зображень обов'язково потрібно збільшувати. Крім цього, на етапі відладки моделі виявився невеликий недолік пов'язаний з roboflow, а саме нерівномірний розподіл зображень на основі класів, так у валідаційний датасет майже не потрапило зображень з нашівкою групи крові. Також система потребує додаткового поліпшення, через збільшення кількості об'єктів, яких вона може виявити, але все це потребує залучення додаткових людей та спеціалістів.

## ВИСНОВКИ

В результаті виконання роботи було створено інтелектуальну систему, яка здатна виявляти на фото-зображеннях та наносити непрозору маску на вразливі об'єкти, тим самим запобігаючи можливому витоку інформації. Для цього було створено датасет з необхідними зображеннями та на його основі навчено модель штучного інтелекту. Для розробки додатку був виконаний пошук та аналіз доступних фреймворків для якомога надійнішої і безпечнішої його реалізації на стороні клієнту, тим самим запобігаючи витоку користувацьких даних чи злам серверу.

Окрім цього було звернено увагу на стрімке зростання цифровізації в наслідок росту технологій та розквіт OSINT розвідки. Також були розглянуті різні системи комп'ютерного зору на основі Deep Learning і обрано найбільш підходящу до задачі модель комп'ютерного зору - YOLO.

За результатами роботи інтелектуальної системи, можна зробити висновок, що застосування розробленого web-додатку знизить ризик витоку чутливої інформації та підвищить рівень обізнаності користувачів у сфері інформаційної гігієни.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. AI computing systems: an application driven perspective / Zidong Du [et al.]. – [S. l.] : Elsevier Science & Technology Books, 2023. – 600 p.
2. Ayyadevara V. K. Pro machine learning algorithms [Electronic resource] / V. Kishore Ayyadevara. – Berkeley, CA : Apress, 2018. – Mode of access: <https://doi.org/10.1007/978-1-4842-3564-5> (date of access: 14.11.2023). – Title from screen.
3. Bellingcat - the home of online investigations [Electronic resource] // [bellingcat.com](http://bellingcat.com). – Mode of access: <https://www.bellingcat.com/app/uploads/2017/10/9OF67tBB15o.jpg> (date of access: 17.11.2023). – Title from screen.
4. Brilakis I. Infrastructure computer vision / Ioannis Brilakis, Carl Thomas Michael Haas. – [S. l.] : Elsevier Science & Technology Books, 2019. – 408 p.
5. Cipolla R. Machine learning for computer vision / Roberto Cipolla. – Berlin, Heidelberg : Springer Berlin Heidelberg, 2013. – 472 p.
6. Davies E. R. Advanced methods and deep learning in computer vision / E. R. Davies, Octavia Camps, Matthew Turk. – [S. l.] : Elsevier Science & Technology Books, 2021. – 550 p.
7. Davies E. R. Computer vision: principles, algorithms, applications, learning / E. R. Davies. – [S. l.] : Elsevier Science & Technology Books, 2017. – 900 p.
8. Freeman L. AI assurance / Laura Freeman, Feras A. Batareseh. – [S. l.] : Elsevier Science & Technology Books, 2022.
9. Huang S.-C. Principles and labs for deep learning / Shih-Chia Huang, Trung-Hieu Le. – [S. l.] : Elsevier Science & Technology, 2021.
10. Ji Q. Probabilistic graphical models for computer vision / Qiang Ji. – [S. l.] : Elsevier Science & Technology Books, 2019. – 294 p.
11. Nilsson N. Quest for artificial intelligence: a history of ideas and achievement / Nilsson Nils J. – [S. l.] : Cambridge University Press, 2013. – 707 p.

12. Karim M. R. TensorFlow: Powerful predictive analytics with TensorFlow : rapid learning solution / Md Rezaul Karim. – Birmingham : Packt Publishing, 2018. – 165 p.
13. Klette R. Performance characterization in computer vision / Reinhard Klette. – Dordrecht : Springer Netherlands, 2000. – 317 p.
14. Scarpino M. TensorFlow / Matthew Scarpino. – Hoboken, New Jersey : John Wiley & Sons, Inc., 2018. – 336 p.
15. Solomon J. Numerical algorithms: methods for computer vision, machine learning, and graphics / Justin Solomon. – Boca Raton : CRC Press, Taylor & Francis Group, 2015. – 379 p.
16. Tanimoto S. Structured computer vision: machine perception through hierarchical computation structures / S. Tanimoto. – [S. l.] : Elsevier Science & Technology Books, 2014. – 246 p.
17. Thomas P. Artificial intelligence / Peggy Thomas. – Farmington Hills, Mich : Lucent Books, 2005. – 111 p.
18. Varonis: we protect data [Electronic resource] // varonis.com. – Mode of access: [https://info.varonis.com/hubfs/Imported\\_Blog\\_Media/VAR\\_OSINTFramework.jpg](https://info.varonis.com/hubfs/Imported_Blog_Media/VAR_OSINTFramework.jpg) (date of access: 20.11.2023). – Title from screen.
19. Zaccone G. Getting Started with TensorFlow / Giancarlo Zaccone. – [S. l.] : Packt Publishing, 2016. – 180 p.
20. YOLOv4: optimal speed and accuracy of object detection [Electronic resource] // arXiv.org. – Mode of access: <https://arxiv.org/abs/2004.10934> (date of access: 17.12.2023). – Title from screen.
21. You only look once: unified, real-time object detection [Electronic resource] // arXiv.org. – Mode of access: <https://arxiv.org/abs/1506.02640> (date of access: 12.11.2023). – Title from screen.
22. Analytics vidhya [Electronic resource] // [www.analyticsvidhya.com](http://www.analyticsvidhya.com). – Mode of access:

- <https://editor.analyticsvidhya.com/uploads/8550388471Untitled%20Ob1.png> (date of access: 15.11.2023). – Title from screen.
- 23.Habr [Electronic resource] // Habr.com. – Mode of access: [https://habrstorage.org/r/w1560/getpro/habr/upload\\_files/359/880/885/35988088553eaa489fdbce15720343e8.png](https://habrstorage.org/r/w1560/getpro/habr/upload_files/359/880/885/35988088553eaa489fdbce15720343e8.png) (date of access: 10.11.2023). – Title from screen.
- 24.MDPI - publisher of open access journals [Electronic resource] // mdpi.com. – Mode of access: [https://www.mdpi.com/electronics/electronics-09-01547/article\\_deploy/html/images/electronics-09-01547-g003.png](https://www.mdpi.com/electronics/electronics-09-01547/article_deploy/html/images/electronics-09-01547-g003.png) (date of access: 13.11.2023). – Title from screen.
- 25.GitHub - ultralytics: YOLOv8 in PyTorch [Electronic resource] // Github.com. – Mode of access: <https://github.com/ultralytics/ultralytics> (date of access: 08.11.2023). – Title from screen.



## ДОДАТОК А

Код навчання моделі використаний в Google Colaboratory

```
# встановлення ultralytics
%pip install ultralytics
import ultralytics
ultralytics.checks()
```

```
#встановлення roboflow для швидкого імпорту датасету через
API
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="BPCLOCYUkBGxZZ4St6MD")
project = rf.workspace("yolo-x4jnz").project("test-yv394")
dataset = project.version(4).download("yolov8")
```

```
#тренування моделі
!yolo task=detect \
mode=train \
model=yolov8s.pt \
data=/content/test-4/data.yaml \
epochs=100 \
imgsz=640
```

```
#валідація моделі
!yolo task=detect \
mode=val \
model=/content/runs/detect/train5/weights/best.pt \
data=/content/test-4/data.yaml
```

```
#перевірка навченої моделі на тренувальному датасеті
!yolo task=detect \
mode=predict \
model=/content/runs/detect/train5/weights/best.pt \
conf=0.25 \
source=/content/test-4/test/images
```

```
#конвертація моделі YOLO в формат який підтримує
TensorFlow.js
from ultralytics import YOLO
```

```
model = YOLO("best.pt")
# Export the model
model.export(format="tfjs")
```

## Код додатку Vue.js

### Файл main.js

```
import { createApp } from 'vue'
import App from './App.vue'

createApp(App).mount('#app')
```

### Файл App.vue

```
<template>
  <div class="App">
    <!-- завантаження -->
    <AppLoader v-if="loading">Loading model... {{
loadingProgress.toFixed(2) }}%</AppLoader>
    <div class="header">
      <h1>📷 Вас вітає детектор вразливих об'єктів на
фото</h1>
      <p>Для початку роботи оберіть необхідне зображення</p>
    </div>

    <div class="content">
      
      <!-- відображення результатів -->
      <canvas :width="model.inputShape[1]"
:height="model.inputShape[2]" ref="canvasRef" />
    </div>

    <!-- опрацювання кнопки-->
    <ButtonHandler @update:imageSrc="handleImageSrcUpdate" />
  </div>
</template>

<script>
import { toRaw } from 'vue';
import * as tf from '@tensorflow/tfjs';
import '@tensorflow/tfjs-backend-webgl';
import AppLoader from './components/AppLoader';
import ButtonHandler from './components/btn-handler';
import { detect } from './components/detect';
```

```

import './app.css';

export default {
  name: 'App',
  components: {
    AppLoader,
    ButtonHandler,
  },
  data() {
    return {
      loading: true,
      loadingProgress: 0,
      model: {
        net: null,
        inputShape: [1, 0, 0, 3],
      },
      modelName: 'yolov8n',
      imageSrc: '#',
      imageVisible: false, // start hide image
    };
  },
  mounted() {
    tf.ready().then(async () => {
      const yolov8 = await
tf.loadGraphModel(`${window.location.href}/${this.modelName}_
web_model/model.json`, {
      onProgress: (fractions) => {
        this.loading = true;
        this.loadingProgress = fractions * 100;
      },
    });
    const dummyInput = tf.ones(yolov8.inputs[0].shape);
    const warmupResults = yolov8.execute(dummyInput);

    this.loading = false;
    this.loadingProgress = 100; // Изменено на 100, так как
загрузка завершена
    this.model.net = yolov8;

    this.model.inputShape = yolov8.inputs[0].shape;

    tf.dispose([warmupResults, dummyInput]);
  });
  },
  methods: {
    handleImageLoad() {
      this.imageVisible = true; // Показываем изображение,
когда оно загружено

```

```

        detect(this.$refs.imageRef, toRaw(this.model),
this.$refs.canvasRef);
    },
    handleImageSrcUpdate(newSrc) {
        this.imageSrc = newSrc;
        this.imageVisible = false; // Скрываем изображение,
        когда источник обновляется
    },

    },
};
</script>

```

### Файл app.css

```

.App {
    height: 100%;
    padding: 0 10px;
    position: relative;
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
}

.App > * {
    margin: 3px 0;
}

.header {
    text-align: center;
}

.header p {
    margin: 5px 0;
}

.code {
    padding: 5px;
    color: greenyellow;
    background-color: black;
    border-radius: 5px;
}

.content > img {
    width: 100%;
    max-width: 720px;
    max-height: 500px;
    border-radius: 10px;
}

```

```
}

.hidden {
  display: none;
}

.visible {
  display: block;
}

.content > video {
  display: none;
  width: 100%;
  max-width: 720px;
  max-height: 500px;
  border-radius: 10px;
}

.content {
  position: relative;
}

.content > canvas {
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
}

button {
  text-decoration: none;
  color: white;
  background-color: black;
  border: 2px solid black;
  margin: 0 5px;
  padding: 5px;
  border-radius: 5px;
  cursor: pointer;
}

button:hover {
  color: black;
  background-color: white;
  border: 2px solid black;
}
```

## Файл Detect.js

```

import * as tf from "@tensorflow/tfjs";
import labels from "./labels.json";

const numClass = labels.length;

const preprocess = (source, modelWidth, modelHeight) => {
  let RatioX, RatioY; // співвідношення

  const input = tf.tidy(() => {
    const img = tf.browser.fromPixels(source);

    // вирівнювання зображення до квадрата => [n, m] to [n, n], n > m
    const [h, w] = img.shape.slice(0, 2); // отримати ширину та висоту джерела
    const maxSize = Math.max(w, h); // отримати максимальний розмір
    const imgPadded = img.pad([
      [0, maxSize - h], // тільки знизу
      [0, maxSize - w], // тільки справа
      [0, 0],
    ]);

    RatioX = maxSize / w; // оновлюємо RatioX
    RatioY = maxSize / h; // оновлюємо RatioY
    console.log(RatioX, RatioY)

    return tf.image
      .resizeBilinear(imgPadded, [modelWidth, modelHeight])
      // змінити розмір фрейму
      .div(255.0) // нормалізація
      .expandDims(0);
  });

  return [input, RatioX, RatioY];
};

const detect = async (source, model, canvasRef, callback = () => {}) => {
  console.log('Detect function called with:', { source, model, canvasRef });
  const [modelWidth, modelHeight] = model.inputShape.slice(1, 3);

  tf.engine().startScope(); // start scoping tf engine

```

```

const [input, RatioX, RatioY] = preprocess(source,
modelWidth, modelHeight); // попередня обробка
const res = model.net.execute(input); // модель виводу
const transRes = res.transpose([0, 2, 1]); // результат
transpose
const boxes = tf.tidy(() => {
  const w = transRes.slice([0, 0, 2], [-1, -1, 1]); //
отримуємо ширину
  const h = transRes.slice([0, 0, 3], [-1, -1, 1]); //
отримуємо висоту
  const x1 = tf.sub(transRes.slice([0, 0, 0], [-1, -1, 1]),
tf.div(w, 2)); // x1
  const y1 = tf.sub(transRes.slice([0, 0, 1], [-1, -1, 1]),
tf.div(h, 2)); // y1
  return tf
    .concat(
      [
        y1,
        x1,
        tf.add(y1, h), //y2
        tf.add(x1, w), //x2
      ],
      2
    )
    .squeeze();
}); // [y1, x1, y2, x2]

const [scores, classes] = tf.tidy(() => {
  // оцінка класу
  const rawScores = transRes.slice([0, 0, 4], [-1, -1,
numClass]).squeeze(0);
  return [rawScores.max(1), rawScores.argmax(1)];
});

const nms = await tf.image.nonMaxSuppressionAsync(boxes,
scores, 500, 0.45, 0.2); // NMS to filter boxes
const boxes_data = boxes.gather(nms, 0).dataSync(); //
індексація боксів
const scores_data = scores.gather(nms, 0).dataSync(); //
індексація балів
const classes_data = classes.gather(nms, 0).dataSync(); //
індексація класів

const ctx = canvasRef.getContext("2d");
ctx.drawImage(source, 0, 0, canvasRef.width,
canvasRef.height);

```

```

    renderBoxes(canvasRef, boxes_data, scores_data,
classes_data, [RatioX, RatioY]); // рендер боксу

    tf.dispose([res, transRes, boxes, scores, classes, nms]);
// очищуємо пам'ять
    callback();

    tf.engine().endScope(); // закриваємо виділену область
};

const renderBoxes = (canvas, boxes_data, scores_data,
classes_data, ratios) => {
    const ctx = canvas.getContext("2d");

    for (let i = 0; i < scores_data.length; ++i) {
        let [y1, x1, y2, x2] = boxes_data.slice(i * 4, (i + 1) *
4);
        x1 *= ratios[0];
        x2 *= ratios[0];
        y1 *= ratios[1];
        y2 *= ratios[1];

        // Замальовуємо знайдений прямокутник
        ctx.fillStyle = "rgba(0, 255, 0, 1)";
        ctx.fillRect(x1, y1, x2 - x1, y2 - y1);
    }
};

export { preprocess, detect };

```

### Файл btn-handler.vue

```

<template>
  <div class="btn-container">
    <!-- Image Handler -->
    <input
      type="file"
      accept="image/*"
      style="display: none"
      @change="handleImageChange"
      ref="inputImageRef"
    />
    <button @click="handleImageClick">
      {{ streaming === 'image' ? 'Close' : 'Open' }} Image
    </button>
  </div>

```



```

</template>

<script>
export default {
  name: 'ButtonHandler',
  data() {
    return {
      streaming: null, // streaming state
    };
  },
  methods: {
    handleImageChange(e) {
      const url = URL.createObjectURL(e.target.files[0]);
      // create blob url
      this.$emit('update:imageSrc', url); // emit event to
update image source
      this.streaming = 'image'; // set streaming to image
    },
    handleImageClick() {
      if (this.streaming === null) {
        this.$refs.inputImageRef.click();
      } else if (this.streaming === 'image') {
        this.closeImage();
      } else {
        alert(`Can't handle more than 1 stream\nCurrently
streaming: ${this.streaming}`);
      }
    },
    closeImage() {
      this.$emit('update:imageSrc', '#'); // emit event to
reset image source
      this.streaming = null; // set streaming to null
    },
  },
};
</script>

```

### Файл AppLoader.vue

```

<template>
  <div class="wrapper" v-bind="$attrs">
    <div class="spinner"></div>
    <p><slot></slot></p>
  </div>
</template>

<script>

```

```
export default {
  name: 'AppLoader',
};
</script>

<style>
.wrapper {
  background-color: rgba(255, 255, 255, 0.5);
  position: absolute;
  width: 100%;
  height: 100%;
  z-index: 1000;
  display: flex;
  justify-content: center;
  align-items: center;
}

.wrapper > .spinner {
  width: 40px;
  height: 40px;
  background-color: #333;

  margin: 10px 10px;
  -webkit-animation: sk-rotateplane 1.2s infinite ease-in-out;
  animation: sk-rotateplane 1.2s infinite ease-in-out;
}

@-webkit-keyframes sk-rotateplane {
  0% {
    -webkit-transform: perspective(120px);
  }
  50% {
    -webkit-transform: perspective(120px) rotateY(180deg);
  }
  100% {
    -webkit-transform: perspective(120px) rotateY(180deg)
    rotateX(180deg);
  }
}

@keyframes sk-rotateplane {
  0% {
    transform: perspective(120px) rotateX(0deg)
    rotateY(0deg);
    -webkit-transform: perspective(120px) rotateX(0deg)
    rotateY(0deg);
  }
}
```

```
50% {
  transform: perspective(120px) rotateX(-180.1deg)
rotateY(0deg);
  -webkit-transform: perspective(120px) rotateX(-180.1deg)
rotateY(0deg);
}
100% {
  transform: perspective(120px) rotateX(-180deg) rotateY(-
179.9deg);
  -webkit-transform: perspective(120px) rotateX(-180deg)
rotateY(-179.9deg);
}
}

.wrapper > p {
  margin: 0;
  font-size: large;
}
</style>
```

Файл labels.json

```
[
  "grup",
  "pozyv",
  "shevr"
]
```