

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Навчально-науковий
Інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра інформаційних технологій та комп'ютерної інженерії
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня магістра

студента Кулікова Артура Віталійовича
(П.І.Б.)

академічної групи 123-22м-1
(шифр)

спеціальності 123 «Комп'ютерна інженерія»
(код і назва спеціальності)

за освітньо-професійною програмою «Комп'ютерна інженерія»
(офіційна назва)

на тему «Обґрунтування параметрів комп'ютерної системи туристичної компанії ТОВ «Лелека»

(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
кваліфікаційної роботи	проф. Цвіркун Л.І.			
розділів:				
теоретичний розділ	проф. Цвіркун Л.І.			
синтез системи	доц. Бешта Д.О.			
розроблення програмного забезпечення	ас. Панферова Я.В.			
Рецензент				
Нормоконтролер	доц. Шедловська Я.І.			

Дніпро
2023

ЗАТВЕРДЖЕНО:
інформаційних технологій
та комп'ютерної інженерії
(повна назва)

_____ Гнатушенко В.В.
(підпис) (прізвище, ініціали)

" ____ " _____ 2023 року.

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня магістра

студенту Куліков А.В. академічної групи 123М-22-1
(прізвище, ініціали) (шифр)

спеціальності 123 Комп'ютерна інженерія
(код і назва спеціальності)

за освітньо-професійною програмою 123 «Комп'ютерна інженерія»
(офіційна назва)

на тему ««Обґрунтування параметрів комп'ютерної системи туристичної компанії ТОВ «Лелека»»»

затверджена наказом ректора НТУ «Дніпровська політехніка» від 09.10.2023 № 1227-с

Розділ	Зміст	Термін виконання
Стан питання та постановка завдання	На основі матеріалів виробничих практик, інших науково-технічних джерел сформулювати наукове завдання, конкретизувати предмет та мету досліджень	10.10.2023
Теоретичний	Обґрунтувати теоретичну базу розв'язання наукового завдання, якому присвячено роботу	25.10.2023
Синтез системи	Розробка комплексу	15.11.2023
Розроблення програмного забезпечення	Розробка програмного забезпечення	29.11.2023
Експериментальний розділ	Проведення і обробка результатів експериментів	06.12.2023

Завдання видано _____ проф. Цвіркун Л.І.
(підпис керівника) (прізвище та ініціали)

Дата видачі 06 вересня 2023 р.

Дата подання до атестаційної комісії 10.12.2023 р.

Прийнято до виконання _____ Куліков А.В.

РЕФЕРАТ

Пояснювальна записка: 72 с., 25 рис., 6 табл., 1 дод., 12 джерел.
СИСТЕМА, ЗАВАНТАЖЕННЯ ЗОБРАЖЕННЯ, ПРОГРАМА,
МОДЕЛЮВАННЯ, ІДЕНТИФІКАЦІЯ

Об'єкт дослідження: комп'ютерна система туристичної компанії ТОВ «Лелека».

Мета роботи: визначення параметрів комп'ютерної системи туристичної компанії ТОВ «Лелека».

У вступі показано актуальність, мету, завдання, об'єкт, предмет, ідею дослідження.

У розділі «Стан питання і постановка задачі» розглянута сфера застосування об'єкта дослідження. Проаналізовано існуючі алгоритми відеоконтролю. Сформульовано мету і завдання роботи.

У теоретичному розділі описані методи обробки інформації, розроблені моделі для перевірки розробленої системи, обґрунтовані параметри апаратно-програмного забезпечення.

У розділі «Синтез системи» проаналізовані складові компоненти комплексу, вибрано обладнання та розроблена структурна схема комп'ютерної системи.

У розділі «Розроблення програмного забезпечення» обоснована структура і методи реалізації програмного забезпечення, розроблено програмне забезпечення для завантаження кадрів зображення з потокового відео камери, знаходження обличчя на вилучених зображеннях, вилучення знайдених обличчя і запис їх в окремий каталог та порівняння записаних обличчя з базами наявних співробітників компанії або небезпечних осіб.

В експериментальному розділі поставлена задача експерименту і обоснована методика його проведення та перевірено розроблене програмне забезпечення.

ЗМІСТ

	Стор.
Перелік скорочень, умовних познач, одиниць і термінів	6
Вступ	7
1 Стан питання і постановка завдання	9
1.1 Огляд галузі застосування комп'ютерної системи	9
1.2 Характеристика об'єкта впровадження	11
1.3 Методи відеоконтролю відвідувачів підприємств	16
1.4 Постановка завдання	17
2 Теоретична частина	18
2.1 Завантаження та обробка зображень для розпізнавання обличчя	27
2.2 Особливості ідентифікації осіб на зображенні	27
2.3 Отримання та тренування моделі для розпізнавання осіб на зображеннях	28
2.4 Розробка моделей відео та фотозображень	30
3 Розробка апаратної частини комп'ютерної системи компанії ТОВ «Лелека»	34
3.1 Вимоги до системи в цілому	34
3.2 Вимоги до функцій комп'ютерної системи компанії «Лелека»	35
3.3 Вимоги до видів забезпечення	36
3.3.1 Вимоги до інформаційного забезпечення	36
3.3.2 Вимоги до лінгвістичного забезпечення	36
3.3.3 Вимоги до технічне забезпечення системи	36
3.4 Вибір обладнання для побудови комп'ютерної системи	39
3.5 Розробка структурної схеми комп'ютерної системи туристичної компанії «Лелека»	42
4 Розроблення програмного забезпечення	43
4.1 Призначення і сфера застосування програми	43
4.2 Обґрунтування технічних характеристик програми	43
4.2.1 Постановка завдання на розробку програми	43
4.2.2 Структура і алгоритм функціонування програми	43
4.2.3 Вибір складу програмних засобів	43
4.3 Опис розробленої програми	44
4.3.1 Загальні відомості	44

4.3.2	Функціональне призначення	44
4.3.2.1	Типи розв'язуваних завдань	44
4.3.2.2	Функціональні обмеження	44
4.3.3	Опис логічної структури програми ідентифікації осіб	44
4.3.4	Використовувані технічні засоби	44
4.3.5	Цикл роботи програми	45
4.3.6	Вхідні та вихідні дані	46
5	Експериментальний розділ	47
5.1	Постановка завдання експерименту і обґрунтування методики	47
5.2	Проведення експерименту для перевірки розробленої системи	47
5.2.1	Перевірка програми по вилученню зображень з потокового відео	47
5.2.2	Перевірка роботи програми по знаходженню обличь осіб на зображеннях та їх вилученню	49
5.2.3	Перевірка роботи програми для ідентифікації осіб	53
5.3	Оцінка результатів	56
5.3.1	Вилучення зображень з потокового відео	56
5.3.2	Знаходженню обличь на зображеннях	57
	Висновки	59
	Перелік посилань	60
	Додаток А. Текст програми	62

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАК, ОДИНИЦЬ І ТЕРМІНІВ

LAN – Local Area Network ;

MAN – Metropolitan Area Network ;

SSL – Secure Socket Layer

SET – Secure Electronic Transactions;

TCP/IP – Transmission Control Protocol/Internet Protocol;

UTP – Unshielded twisted pair ;

WAN – Wide Area Network

ЕОМ – електронна обчислювальна машина;

КІСП – комп'ютерна інформаційна система підприємства ;

ЛОМ – локальна обчислювальна мережа;

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер ;

ТОВ – товариство з обмеженою відповідальністю .

ВСТУП

Інфраструктура інформаційних технологій підприємства, головним чином, базується на локальній обчислювальній мережі, тому, наскільки якісно буде спроектована і розроблена ця мережа, залежатимуть показники якості функціонування інфраструктури в цілому. Локальна обчислювальна мережа – це ІТ-система, що за допомогою техніки забезпечує безперебійне інформаційне поле для вирішення бізнес-завдань [1].

Програмне забезпечення, термінальні підключення, бази даних, що потребує реалізації стабільного та якісного з'єднання, змушують приділяти особливу увагу при виборі обладнання та програмного забезпечення. Ця вимога справедлива як усередині корпоративної мережі, так і для доступу до глобальної мережі.

Метою кваліфікаційної роботи є вивчення особливостей отримання зображень осіб, які входять до приміщення компанії «Лелека», дослідження процесу та визначення параметрів вилучення фрагментів з обличчями та формування баз осіб з різними видами дозволів доступу до приміщень.

Мета і завдання дослідження: *Метою роботи* є обґрунтування параметрів комп'ютерної системи туристичної компанії «Лелека» з розробкою програмно-апаратного комплексу відео контролю доступу до приміщень.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз галузі та умов застосування системи та огляд об'єкта впровадження;
- виконати аналіз особливостей запису кадрів зображення з потокового відео камери;
- знаходження обличчя на вилучених зображеннях мово Python з бібліотекою;
- вилучення знайдених обличчя і запис їх в окремий каталог;
- порівняння записаних обличчя з базами наявних співробітників компанії або небезпечних осіб;

– передача інформації про розпізнання відповідному співробітнику компанії «Лелека»;

– обґрунтувати необхідність застосування камер для контролю осіб, які входять до приміщення компанії;

– провести експериментальні дослідження і обґрунтувати параметри комп'ютерної системи контролю туристичної компанії «Лелека».

Об'єкт дослідження: комп'ютерна система туристичної компанії «Лелека».

Предмет дослідження: підготовки зображень для розпізнавання осіб.

Методи дослідження: теорія нейромереж, теорія моделювання.

Ідея роботи: додаткове обладнання комп'ютерної системи комплексом контролю з розпізнаванням осіб по обличчю для уникнення несанкційованого входу на територію компанії.

Наукові положення:

1. Встановлено, що для виявлення обличчя на зображеннях на вході в компанію «Лелека» необхідно змонтувати камеру на відстані до 5 м та встановити додаткове освітлення, що підвищує кількість знайдених обличчя на зображенні.

Наукові результати:

1. Обґрунтовано використання бібліотеки Face Recognition, робота якої базується на нейромережі dlib, з достовірністю понад 0.9938 для розпізнавання осіб по обличчю на вході в приміщення туристичної компанії «Лелека».

Обґрунтованість і достовірність наукових положень, висновків і рекомендацій підтверджуються результатами експериментальних досліджень.

Практичне значення отриманих результатів полягає в розробці програмного забезпечення, яке дозволяє для розпізнавання осіб завантажувати його з відеокамери, виявляти на ньому обличчя і збирати їх зображення в окрему базу.

1 СТАН ПИТАННЯ І ПОСТАНОВКА ЗАВДАННЯ

1.1 Огляд галузі застосування комп'ютерної системи

Туристичний бізнес, є однією з найбільш динамічних сфер економіки в мирний час, а також високо насиченою інформаційною сферою діяльності. Таким чином, збирання, зберігання, обробка та надання актуальної інформації є важливою та необхідною умовою функціонування будь-якого туристичного підприємства. Не так багато сфер економіки, в яких успіх бізнесу може залежати від швидкості передачі та прийому інформації, актуальності та своєчасності отримання, від її адекватності та повноти. Зважаючи на ці фактори, успішний розвиток туристичного підприємства передбачає активне використання нових технологій як у сфері створення туристичних продуктів, так і їх просування.

Сучасні інформаційні технології займають одну з найважливіших позицій у сфері туристичного бізнесу. Бізнес туризму потребує використання комп'ютерних технологій.

Сьогодні у сфері туризму використовують досить багато нових комп'ютерних технологій, таких як глобальні комп'ютерні системи резервування (GDS), інтегровані комунікаційні мережі, мультимедійні системи, Smart Cards, інформаційні системи менеджменту та інше. Перелічені вище інформаційні технології застосовуються з різною частотою та мають неоднакове поширення. Відрізняється також рівень їхнього впливу на розвиток туристичної промисловості.

Структура туристичної індустрії подібна до інших сфер економічної діяльності. Важливою властивістю туристичної галузі є те, що вона є єдиним центром, який утримує різних постачальників послуг у межах туристичної галузі. Інформаційні потоки забезпечують зв'язок між постачальниками та споживачами туристичних послуг, причому вони йдуть як у вигляді потоків даних, так і у формі послуг та платежів.

Інформаційних технологій відіграють надважливу роль у галузі туризму, про це свідчить той факт, що туризм входить до переліку галузей з найбільшою реалізацією товарів та послуг через мережу Internet. За оцінками провідних світових експертів, перспективи, туристична галузь стане першою щодо застосування інформаційних технологій та реалізації своїх послуг через глобальну мережу.

Переваги використання мережі Internet для суб'єктів туристичного ринку України очевидні: вона дає найповніший доступ до інформації, дозволяє заощадити кошти під час комунікації із закордонними постачальниками послуг, надає інформацію про передові технології, дозволяє обмінюватися відгуками та інформацією з колегами. Слід зазначити, що надання туристичних послуг у мережі Internet є дешевшим, ніж звичайно, а найчастіше ефективніше, оскільки направлено на певну цільову аудиторію. Це дуже важливо для постачальників туристичних послуг в Україні, які займаються в'їзним міжнародним туризмом тому, що це вливання коштів в Українську економіку, а також популяризація найцікавіших місць та міст України серед іноземних громадян.

1.2 Характеристика об'єкта впровадження

Сучасна туристична фірма надає дуже широкий спектр послуг своїм клієнтам. Серед клієнтів не тільки фізичні особи, але і організації та великі підприємства.

Коректне відношення до потреб клієнтів – основа успіху в бізнесі.

Успішна діяльність будь-якого підприємства багато в чому залежить не тільки від керуючого компанією, а й від персоналу, а щоб він добре працював, необхідно подумати про гарне оснащення приміщення.

Приміщення буде поділено на 2 кімнати: в одній кімнаті будуть керівники, а в іншій робочий персонал.

До посадових обов'язків працівників туристичної компанії ТОВ «Лелека» належить.

Директор: розробка іміджу фірми, взаємодія з партнерами, маркетинг, представницькі функції, складання програм перебування туристів за кордоном, ділове листування.

Головний менеджер: оперативне управління, взаємодія з транспортними фірмами, організація проживання та харчування туристів, юридичні аспекти діяльності фірми.

Менеджери з напрямків: робота з клієнтами, оформлення договорів із клієнтами на туристське обслуговування.

Бухгалтер: робота із фінансовими справами фірми, ведення книг записів.

Кур'єр: доставка різного роду документації.

Діюча організаційна структура управління туристичної компанії ТОВ «Лелека» представлена на рисунку 1.1. Організаційна структура підприємства належить до класичної лінійної структури.

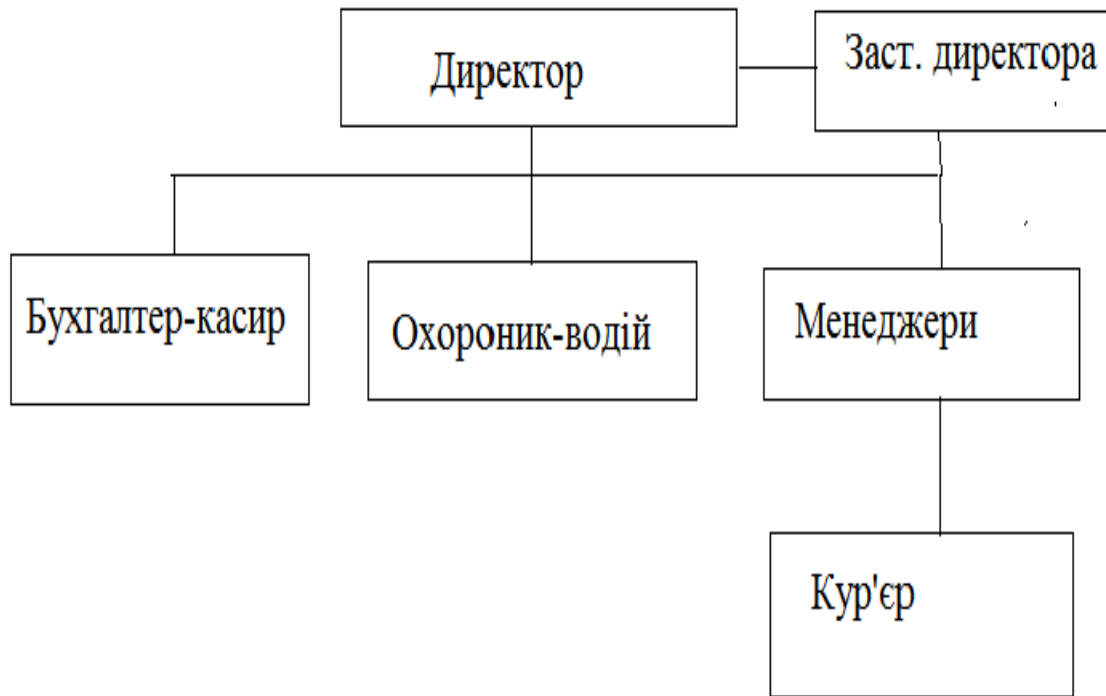


Рисунок 1.1 – Організаційна структура туристичної компанії ТОВ «Лелека»

Право на доступ до інформаційних ресурсів підприємства розмежовувань, відповідно до посади кожного користувача, що забезпечується системою паролів. Комп'ютерна мережа є розподіленою. Також у компанії є віддалене відділення по роботі з клієнтами, тому чималу увагу треба приділяти захисту даних при з'єднання з мережею цього відділення.

Головне відділення компанії ТОВ «Лелека» розміщено в двоповерховому офісному будинку. Розміри будівлі – 20*12 м. Також компанія має віддалене відділення по роботі з клієнтами, яке розташоване на першому поверсі 9 поверхової будівлі.

Приміщення компанії має такі розміри (рисунок 1.2, 1.3):

Кабінет директора – 8*7 м;

Коридор 1 поверху – 8*4 м;

Відділ менеджменту – 9*8 м;

Відділ роботи з клієнтами – 14*4 м;

Санвузол – 6*4 м;

Кабінет головного менеджера – 12*7 м;

Коридор 2 поверхи – 7*4 м;

Бухгалтерія-9*7 м;

Відділ керування персоналом 13*5 м.



Рисунок 1.2 – Генеральний план 1 поверху офісної будівлі



Рисунок 1.3 – Генеральний план 2 поверху офісної будівлі

Також компанія має віддалене відділення по роботі з клієнтами, яке розташоване на першому поверсі багатоповерхового будинку.

Основний вид діяльності підприємства – організація продажу сформованих туроператором турів.

Також ТОВ «Лелека» займається просуванням та реалізацією туристських продуктів, що надаються операторами ринку туристичних послуг, на комісійній основі з метою отримання прибутку. При цьому підприємство має право розміщувати рекламу, брати участь у спеціалізованих виставках та ярмарках, видавати відповідну поліграфічну продукцію та організувати центр з продажу туристичних продуктів.

Клієнтами ТОВ «Лелека» є громадяни України переважно мешканці міста Дніпро. Агентство надає послуги з туризму внутрішнього та туризму виїзного.

Категорії туристичних поїздок:

Короткі 1-2 дні або 4-7 днів;

Середні 8-28 днів;

Довгі 29-91 день.

Послуги, що пропонуються фірмою, включають:

- підготовку проїзних документів та віз в іноземних посольствах;
- заказ для клієнтів авіаквитків та відповідного трансфера;
- забезпечення клієнтів поселенням в готелях з необхідними умовами для відпочинку та роботи;
- заказ для клієнтів різних екскурсій та розваг, забезпечення супроводжуваними та гідями: надання можливостей для проведення ділових зустрічей та наукових конференцій комп'ютерами тощо). д. а також можливість користуватися послугами перекладачів та секретарів);
- організацію для своїх клієнтів туристичні поїздки з різних напрямків.

У структурі та реалізації підприємства у 2021-2022 році на внутрішній туризм припадає 28,6%, відповідно на візовий туризм – 71,4%.

Основні напрямки візового туризму: Туреччина, Іспанія, Греція, Болгарія, Кіпр, Італія та ін. Внутрішнього туризму: Карпати, Кам'янець-Подільський, Чернівці, Ужгород, Мукачеве, Умань.

Спектр послуг компанії ТОВ «Лелека» з організації:

- бронювання та оформлення авіаквитків;
- оформлення квитків на потяги та автобусні рейсові лінії;
- бронювання готелів в Україні та по всьому світу;
- групових та індивідуальних бізнес-поїздок будь-якої складності;
- відпочинкових та екскурсійних турів у будь-які країни;
- дитячих таборів в Україні та за кордоном;
- візової підтримки та послуг страхування;
- надати клієнтам весь спектр послуг туристичного ринку;
- забезпечити високий рівень обслуговування за єдиним стандартом якості;
- зберегти індивідуальний підхід кожного клієнта.

Компанія буде приділяти особливу увагу формуванню спеціальних пакетів, до яких включені, розроблені відповідно до побажання клієнта, пропозиції щодо організації бізнес-туризму, індивідуальних та групових поїздок. Що включає наступне:

- бронювання номерів у готелях та авіаквитків;
- транспортні маршрути будь-якої складності, трансфери;
- VIP-обслуговування;
- екскурсійне обслуговування, послуги гідів, перекладачів;
- сприяння в оформленні закордонного паспорта;
- візова підтримка, оформлення страхового полісу;
- продаж турів у кредит.

Туристична компанія має за головну мету:

- забезпечення своїх клієнтів якісними послугами,
- залучення до співпраці незалежних інвесторів та кредиторів,
- співпраця з підприємствами сфери туризму регіону.

1.3 Методи відеоконтролю відвідувачів підприємств

Одним з методів із запобігання появи небажаних відвідувачів в приміщеннях підприємств є встановлення відеоспостереження на вході. Встановлення найсучасніших IP камер значно розширює можливості таких систем.

Системи відеоспостереження необхідно доповнити підсистемою розпізнавання осіб по зображенням. Такий підхід застосовується під час дистанційної ідентифікації клієнтів банків, страхувальних компаній, для верифікації в логістиці, авіакомпаніях, торгівлі, лікарнях тощо.

Нейронні мережі допомагають проводити це в реальному часі.

Можна виділити декілька кроків в такому процесі розпізнавання осіб по зображенням:

- завантаження зображення в систему;
- обробка отриманих даних. На цьому кроці можливе покращення зображення за рахунок підвищення контрастності, деталізації та насиченості кольорів, видалення спотворених ділянок зображення, з шумом або розмитих. Іноді рекомендується переводити зображення у чорно-білий формат;
- виділення з зображення обличчя і розміщення в окремій фотогалереї;
- порівняння отриманих зображення обличчя з зображеннями небажаних відвідувачів або злочинців.

1.4 Постановка завдання

Основним завданням кваліфікаційної роботи є створення комп'ютерної системи туристичної компанії ТОВ «Лелека» з опрацюванням структурної схеми.

Комп'ютерної системи повинна складатися з 5 підсистем.

Топологія комп'ютерної системи повинна забезпечувати функціонування підприємства з урахуванням його організації і плану розміщення підрозділів в приміщеннях.

Для забезпечення роботи сучасних інформаційних технологій туристичної фірми необхідно використовувати стандарт 1000BASE-T.

Для безпечної роботи туристичної компанії ТОВ «Лелека» система додатково облаштовується модулем відеоконтролю на вході в приміщення з метою оповіщення про появу небезпечних відвідувачів.

Фотогалерея небажаних відвідувачів формується як співробітниками туристичної компанії ТОВ «Лелека», так за допомогою оповіщення від інших компаній та поліції.

2 ТЕОРЕТИЧНА ЧАСТИНА

2.1 Завантаження та обробка зображень для розпізнавання обличчя

Обробка зображень передбачає перетворення зображень в цифрову форму для виконання над ними подальших операцій.

Для виконання розпізнавання осіб по зображенням їх необхідно спочатку завантажити в комп'ютерну систему.

Розпізнавання осіб на зображеннях виконується за допомогою програм написаних поширеною для цього мовою Python.

Для виконання такої складної операції застосовується ряд бібліотек для обробки зображень мовою Python.

Розглянемо ці бібліотеки.

OpenCV – ця бібліотека розроблена в 2000 році. Вона дозволяє виявляти на зображеннях об'єктів, обличь, виконувати сегментацію зображень, розпізнавання осіб тощо. OpenCV має декілька інтерфейсів, широкий доступ до алгоритмів та може застосовуватись сумісно з NumPy, SciPy и Matplotlib.

Для пошуку обличь на зображенні використовується принцип «вікна, що ковзає по зображенню» (метод Віоли-Джонса). Для кожної області зображення, над якою проходить вікно, обчислюється атрибут Хаара. Наявність або відсутність об'єкта у вікні визначається різницею між значенням ознаки та навченим порогом. Вікно проходить по всьому зображенню.

Для пошуку граней більшого масштабу після кожного циклу проходження зображення розмір вікна збільшується.

Тест Хаара складається з порівняння суміжних прямокутних ділянок зображення. Вони розташовуються на зображенні, потім інтенсивності пікселів у областях підсумовуються, після чого обчислюється різниця між сумами.

Ця різниця буде значенням певної ознаки (розміру), певним чином розташованої на зображенні.

Спільним для всіх зображень обличчя є те, що область навколо очей темніша за область навколо щік. Таким чином, загальною ознакою Хаара для обличчя є наявність двох суміжних прямокутних областей над очима та щоками.

Pillow – ця бібліотека розширена версія PIL (Python Imaging Library). Вона є однією з найкращих для обробки зображень. Pillow проста у використанні, підтримує різні методи обробки зображень та різні формати зображень (JPEG, PNG).

Face Recognition Library – ця бібліотека має високу точність, проста у використанні, може бути легко встановлена на сервері без дорогого графічного процесора та має досить прості вимоги. Крім того у порівнянні з OpenCV при пошуку обличь спрацьовує і при обертанні голови більше ніж на 30 градусів. Для ідентифікації особи по координатам характерних точок та їх співвідношенням складається карта обличчя з 128 параметрами. Створення баз таких карт дозволяє розпізнавати осіб на зображеннях.

NumPy – ця бібліотека застосовується для чисельного аналізу, має матрицю та багатовимірні масиви як структури даних. NumPy має високошвидкісну обробку масивів, компактне сховище даних та сумісність даних з іншими бібліотеками.

Matplotlib – ця бібліотека застосовується для обробки зображень і має два методи для читання і відображення. Matplotlib може надавати високоякісні зображення та графіки у різних форматах, має відкритий код і проста та зручна у використанні.

Зображення осіб можна отримувати: копіюючи їх з смартфона, з Інтернету, скануючи вже готові фотографії або з відеокамер. Відеокамери можуть мати режим роботи, коли вони надсилають окремі зображення через заданий період часу. Але більш універсальний спосіб, який підходить і для розробляємої комп'ютерної системи туристичної компанії ТОВ «Лелека» це отримання з камери потокового відео, обробка його з відокремлюванням фреймів і запис їх окремими файлами як зображення.

Розглянемо як це зробити із застосуванням перерахованих бібліотек і мови Python з записаного відео-файлу, а потім при підключенні до відео камери.

Середовище для роботи програми включатиме каталог для вихідних відео-файлів (video) та каталог (video-img) із зображеннями, які отримані з відео-файлів або відеокамери. В програмі будуть задіяні бібліотеки OpenCV, NumPy, os, pathlib.

Створимо програму в якій завантажимо відео-файл у змінну cap і організуємо цикл працюючий продовж виводу відео while (cap.isOpened()). В циклу визначаємо параметри (fps, divider) зчитування фреймів і запису файлів із зображеннями та номера фрейму (frame_id).

Якщо залишок від поділу номера фрейму (frame_id) на дільник (divider) дорівнює нулю черговий фрейм записується у файл як зображення.

```
def take_img_from_video(video_path):
    cap = cv.VideoCapture(video_path)
    wave = Path(video_path)
    count = 0

    if not os.path.exists("video_img"):
        os.mkdir("video_img")

    while (cap.isOpened()):
        ret, frame = cap.read()
        fps = cap.get(cv.CAP_PROP_FPS)
        divider = int(fps * 5)

        if ret:
            frame_id = int(round(cap.get(1)))
            cv.imshow("Frame", frame)
            k = cv.waitKey(2)

            if int(frame_id % divider) == 0:
```

```
cv.imwrite(f"video_img/{count}_{wave.name}.jpg", frame)
print(f"Take a screenshot {count}_{wave.name}")
count += 1
```

Крім цього організуємо поза цикловий запис фрейму при натисканні клавіші «пробіл», завершення роботи програми – при натисканні клавіші «q» та вивід відповідних повідомлень на дисплей.

```
if k == ord(" "):
    cv.imwrite(f"video_img/{count}_br_{wave.name}.jpg", frame)
    print(f"Take a screenshot br_{count}_{wave.name}")
    count += 1

if k == ord("q"):
    print(f"Q pressed, end of the app")
    break

else:
    print("It's end!")
    break
```

Далі звільняємо пам'ять зайняту змінною cap, закриваємо всі відчинені вікна Windows.

```
cap.release()
cv.destroyAllWindows()
```

2.2 Особливості ідентифікації осіб на зображенні

Аналіз бібліотек мови Python показав, що найкращими є бібліотеки для обробки зображень Face Recognition Library, Pillow та OpenCV, а для чисельного аналізу з компактним сховищем даних і високошвидкісною обробкою масивів – бібліотека NumPy.

Розглянемо їх застосування для ідентифікації осіб на зображенні.

Спочатку створимо середовище для роботи програми. Воно повинно включати каталог для вихідних зображень (img) та каталог із зображеннями на яких знайдено і виділено обличчя осіб (img_upd).

Імпортуємо необхідні бібліотеки, визначимо шляхи до зображень перевіримо наявність каталогів img та img_upd. Якщо каталогу img нема, виведемо на дисплей повідомлення про помилку, а якщо каталогу img_upd нема, тоді створимо його.

```
import os
import face_recognition
from PIL import Image, ImageDraw

def face_rec():
    file1 = "img/dipl1.jpg"
    file1_n = "img_upd/dipl1_n.jpg"

    file2 = "img/konfl.jpg"
    file2_n = "img_upd/konfl_n.jpg"

    if not os.path.exists("img"):
        print("[ERROR] there is no directory 'img'")

    if not os.path.exists("img_upd"):
        print("New directory img_upd created!")
        os.makedirs('img_upd')
```

За допомогою бібліотеки `face_recognition` завантажимо два файли (`file1` та `file2`) і спробуємо визначити наявність обличь на зображенні. Координати визначених зображень занесемо в масиви `img1_location` та `img2_location` і виведемо їх на дисплей та кількість знайдених обличь.

```
img1 = face_recognition.load_image_file(file1)
img1_location = face_recognition.face_locations(img1)

img2 = face_recognition.load_image_file(file2)
img2_location = face_recognition.face_locations(img2)

print(img1_location)
print(img2_location)
print(f'Found {len(img1_location)} face(s) in this image')
print(f'Found {len(img2_location)} face(s) in this image')
```

Із застосуванням бібліотеки `PIL` по визначеним координатам сформуємо прямокутники і нанесемо їх на зображення `pil_img1` та `pil_img2` та зберігаємо у файлах (`file1_n`, `file2_2`).

```
pil_img1 = Image.fromarray(img1)
draw1 = ImageDraw.Draw(pil_img1)

for(top, right, bottom, left) in img1_location:
    draw1.rectangle(((left, top), (right, bottom)), outline=(255, 255, 0), width=3)

del draw1
pil_img1.save(file1_n)

pil_img2 = Image.fromarray(img2)
draw2 = ImageDraw.Draw(pil_img2)
```

```

for(top, righ, bottom, left) in img2_location:
    draw2.rectangle(((left, top), (righ, bottom)), outline=(255, 255, 0), width=3)

del draw2

pil_img2.save(file2_n)

```

Далі необхідно вилучити з зображення обличчя обличчя і розмістити їх у вигляді окремих файлів в окремому каталозі-фотогалереї, наприклад `img_faces`. Це може бути фотогалерея небезпечних осіб або групи співробітників компанії ТОВ «Лелека», якщо ставиться задача надання обмеженого доступу до окремих приміщень з відеоконтролем по обличчю.

Створимо відповідну функцію.

Спочатку перевіримо наявність каталогу `img_faces`. Якщо каталогу `img_faces` нема, тоді створимо його.

Для вилучення використаємо систему розпізнавання обличчя на основі глибокого навчання за допомогою бібліотеки `face_recognition`.

Спочатку знаходимо розташування обличчя на зображенні, а потім організуємо цикл і послідовно зчитуємо знайдені обличчя на зображенні та їх запис в каталог `img_faces`. Кожен файл має назву, яка складається з показання створеного лічильника і назви похідного файлу.

В кінці роботи на дисплей виводимо повідомлення про кількість знайдених обличчя на вихідному зображенні.

```

def extracting_faces(img_path):
    if not os.path.exists("img"):
        print("[ERROR] there is no directory 'img'")

    if not os.path.exists("img_faces"):
        print("New directory img_faces created!")
        os.makedirs('img_faces')

    faces = face_recognition.load_image_file(img_path)

```



```

faces_locations = face_recognition.face_locations(faces)

for face_location in faces_locations:
    top, right, bottom, left = face_location

    face_img = faces[top:bottom, left:right]
    pil_img = Image.fromarray(face_img)
    pil_img.save(f"img_faces/{count}_{wave.name}")
    count += 1
    #print(count)

    print (f"Found {count} face(s) in this photo")

```

Таким чином формується набір зображень небезпечних осіб з яким можливо виконувати порівняння і контролювати доступ в приміщення туристичної компанії ТОВ «Лелека».

При отриманні нового зображення особи, яка відвідує туристичну компанію ТОВ «Лелека», необхідно виконати порівняння з наявними в комп'ютерній системі зображеннями небезпечних осіб.

Створимо відповідну функцію для порівняння двох зображень.

```

def compare_faces(img1_path, img2_path):
    img1 = face_recognition.load_image_file(img1_path)
    img1_encodings = face_recognition.face_encodings(img1)[0]
    print(img1_encodings)

    img2 = face_recognition.load_image_file(img2_path)
    img2_encodings = face_recognition.face_encodings(img2)[0]
    print(img1_encodings)

```

```
result = face_recognition.compare_faces([img1_encodings],  
img2_encodings)  
print(result)
```

Якщо це співробітник туристичної компанії ТОВ «Лелека», то видається запрошення на вхід, якщо ні – пропонується почекати менеджера.

```
if result[0]:  
    print("Welcome to the Travel company Leleka!")  
else:  
    print("Sorry, Sorry, wait for the manager!")
```

2.3 Отримання та тренування моделі для розпізнавання осіб на зображеннях

Моделі зображень осіб в бібліотеці `face_recognition` мають 128 параметрів і для підвищення точності розпізнавання потребують тренуванню.

Розробимо для цього наступну функцію.

Цифрові зображення осіб, які підлягають тренуванню, будемо зберігати в каталозі `dataset`. Для кожної особи, яка заноситься в базу системи необхідно підібрати фото з різним виразами обличчя. Чим більше таких фотографій тим краще, тому що це буде покращувати роботу в подальшому.

Далі треба перевірити наявність каталогу `dataset`, організувати базу даних `known_encoding` і отримати перелік файлів в каталозі `dataset`.

```
def train_model_by_img(name):
    if not os.path.exists("dataset"):
        print("[ERROR] there is no directory 'dataset'")
        sys.exit()

    known_encodings = []
    images = os.listdir("dataset")
```

Далі організуємо цикл з перевірки зображень у каталозі `dataset` і порівняння з базовим зображенням. Якщо це одна особа на обох зображеннях, то отримані дані моделі заносяться до бази `face_enc`. Якщо ні – виводиться про це повідомлення і завантажуються нове зображення.

```
for(i, image) in enumerate(images):
    print(f"[+] procesing img {i + 1}/{len(images)}")
    print(image)

    face_img = face_recognition.load_image_file(f'dataset/{image}')
```

```

face_enc = face_recognition.face_encodings(face_img)[0]

#print(face_enc)

if len(known_encodings) == 0:
    known_encodings.append(face_enc)
else:
    for item in range(0, len(known_encodings)):
        result = face_recognition.compare_faces([face_enc],
known_encodings[item])
        print(result)

        if result[0]:
            known_encodings.append(face_enc)
            print("Same person!")
            break
        else:
            print("Another person!")
            break

print(known_encodings)
print(f'Lengh {len(known_encodings)}')

```

Далі збережемо отримані коди моделі заданої особи. Для цього створимо словник, в якому в першій парі буде прізвище особи name та перелік розпізнаних кодувань encodings. Перетворимо отримані дані в потік двійкових даних і запишемо у файл {name}_encodings.pickle.

```

data = {
    "name": name,
    "encodings": known_encodings

```

```
}
```

```
with open(f'{name}_encodings.pickle', "wb") as file:
```

```
    file.write(pickle.dumps(data))
```

```
return f"[INFO] File {name}_encodings.pickle successfully created"
```

2.4 Розробка моделей відео та фотозображень

Щоб провести експериментальну перевірку розробленої комп'ютерної системи проведемо підготовку моделей зображень осіб.

Для перевірки системи необхідні такі похідні моделі:

- відео з особою або з декількома особами для перевірки програми по вилученню зображень з потокового відео;
- фотографії особи або різних осіб з різними положеннями голови, різними виразами обличчя (посмішка, відкритий рот, прикриті очі) для перевірки знаходження обличчя на зображенні та їх вилучення в окремий каталог. Можливо для цього застосувати зображення осіб, які вилучені з відео і відповідають перерахованим вимогам. Також для цього можливо застосувати фотографії відкритих баз даних в Інтернеті;
- фотографії з великою кількістю осіб, які отримані під час проведення тематичних зборів, конференцій, різних зібрань тощо теж для перевірки знаходження обличчя на зображенні та їх вилучення в окремий каталог. За рахунок великої кількості при фотографуванні особи мають різноманітні вирази обличчя, положення голови, що відповідає вимогам перевірки;
- фотографії другої групи, якщо їх зробити для однієї людини, можна застосувати для тренування моделі.

Відповідно до опису підберемо похідні моделі з характеристиками показаними в таблиці 2.1, де модель №1 – відео з особами (на рисунку 2.1 показано фрагмент відео), модель №2 – фотографії особи з різними положеннями голови (рисунок 2.2), модель №3, №4 – фотографії конференції.

Таблиця 2.1 – Характеристики похідних моделей

№ моделі	Рисунок	Формат файлу	Розмір файлу, Кбайт	Розмір зображення, пкс	Роздільна здатність, точок/дюйм
1	2.1	mp4	59 200	1280 x 720	96
2	2.2	jpg	145	520 x 600	120
3	2.3	jpg	593	2048 x 1365	96
4	2.4	jpg	476	1536 x 1024	96



Рисунок 2.1 – Фрагмент відео-файлу, модель №1 для тестування програми



Рисунок 2.2 – Модель зображення №2 для тестування програми



Рисунок 2.3 – Модель зображення №3 для тестування програми



Рисунок 2.4 – Модель зображення №4 для тестування програми

3 РОЗРОБКА АПАРАТНОЇ ЧАСТИНИ КОМП'ЮТЕРНОЇ СИСТЕМИ КОМПАНІЇ ТОВ «ЛЕЛЕКА»

3.1 Вимоги до системи в цілому

Комп'ютерна система компанії ТОВ «Лелека» повинна забезпечувати наступне:

а) збір, обробку та зберігання інформації:

- 1) про технічні параметри встановлених інформаційних засобів;
- 2) про тарифи на універсальні послуги зв'язку;
- 3) про обсяги прийнятої та переданої інформації;
- 4) про обсяги наданих універсальних послуг, виражених у хвилинах або мегабайтах, а також кількість встановлених з'єднань (сеансів зв'язку);

б) облік та передачу в інформаційні засоби діючих ключів аутентифікації;

в) захист від несанкціонованого доступу до інформації, що міститься в системі управління, моніторингу та контролю за технічним станом інформаційних засобів;

г) формування звітів про надані послуги з передачі даних та надання доступу до інформаційно-телекомунікаційної мережі "Інтернет" та звітів про технічний стан інформаційних засобів;

д) виявлення та інформування оператора про появлення осіб, які занесені в базу небажаних та злочинців, формування звітів про них, а також надання статистичної інформації, необхідної оператору для аналізу роботи інформаційних засобів та вживання своєчасних заходів щодо усунення виявлених пошкоджень.

3.2 Вимоги до функцій комп'ютерної системи компанії «Лелека»

Комп'ютерна система компанії «Лелека» повинна мати комплекс відеоконтролю відвідувача або співробітника після входу в приміщення компанії «Лелека» з наступними функціями:

- визначення руху на вході в приміщення в компанію «Лелека»;
- вилученню зображень з потокового відео камери;
- знаходження обличь на вилучених зображеннях;
- вилучення знайдених обличь і запис їх в окремий каталог;
- порівняння записаних обличь з базами наявних співробітників компанії або небезпечних осіб;
- передача інформації про розпізнання відповідному співробітнику компанії «Лелека»;
- тренування нейромережі dpl та доповнення бази наявних співробітників компанії або небезпечних осіб новими моделями.

Активне і пасивне обладнання компанії «Лелека» повинно працювати не менше 12 годин на добу (час на виконання ремонту, налаштування, регламентних робіт не враховується).

Підрозділи компанії мають зв'язок один з іншим, а також між співробітниками за рахунок системи ієрархічного доступу.

3.3 Вимоги до видів забезпечення

3.3.1 Вимоги до інформаційного забезпечення

Однією з характерних особливостей туристичної діяльності є велика кількість та різноманітність інформаційних потоків, що супроводжується їх постійною актуалізацією та високою швидкістю обмінних операцій.

Інформаційні взаємозв'язки існують між усіма учасниками туристського ринку, при цьому особливе значення надається інформації, призначеної для кінцевих користувачів (туристів). Це зумовлено специфічними рисами туристського продукту та необхідністю надання максимально вичерпної інформації для потенційних туристів.

Якість інформаційного забезпечення туризму є найважливішим фактором, який безпосередньо впливає на якість туристських товарів та послуг. Виходячи з цього, під час підготовки кадрів для сфери туризму особливе значення має бути надано інформаційній складовій, зокрема, інформаційне забезпечення туризму.

3.3.2 Вимоги до лінгвістичного забезпечення

Розробка програмного забезпечення треба виконати мовою Python із застосуванням бібліотек OpenCv, NumPy, Face Recognition та інших.

Інтерфейс програм - українською або англійською мовами.

3.3.3 Вимоги до технічне забезпечення системи

При виборі мережного обладнання необхідно враховувати:

- рівень уніфікації обладнання та його сумісність із найбільш поширеним програмним забезпеченням;
- швидкість передачі інформації та перспективи її подальшого збільшення;
- топології мережі та їх можливі комбінації;
- метод управління обміном даними у мережі (CSMA/CD, повний дуплекс або маркерний метод);

- типи кабелю мережі та його основні характеристики;
- технічні характеристики та вартість апаратних засобів (мережевих адаптерів, трансиверів, репітерів, концентраторів, комутаторів).

Вимоги до технічного забезпечення представлені в таблицях 3.1 та 3.2.

Таблиця 3.1 – Технічні вимоги до маршрутизатора

Тип	Характеристика
Процесор	ARM, не менш ніж, 680MHz
Memory	Не менш ніж: 512MB DDR
Hard Disk	Не менш ніж: 512MB на чипі пам'яті NAND, microSD слот
Ethernet порти	Не менш ніж: П'яти 10/100/1000 Mbit/s Ethernet портів з підтримкою Auto MDI/X
Працездатність у режимі брандмауера	Не менш ніж 1 Гбіт/с
Підтримка протоколів маршрутизації RIP, OSPF, BGP	Так
Підтримка EoIP тунелів	Необмежено
Підтримка PPPoE тунелів	Не менш ніж 500
Підтримка VLAN інтерфейсів	Необмежено
Підтримка PPTP тунелів	Не менш ніж 500
Підтримка OVPN тунелів	Необмежено
Підтримка L2TP тунелів	Не менш ніж 500
Активних користувачів Хот-Спот	500
Правила брандмауера P2P	Необмежено
NAT правила	Необмежено

Таблиця 3.2 – Технічні вимоги до комутатора

Тип	Характеристика
Портів Gigabit Ethernet 10/100/1000 (кількість)	8-24 порти
Кількість портів SFP	Від 4 слот
Пропускна здатність	Від 48 Гбіт/сек; 35.7 Mpps
Об'єм буфера пакетів	до 0.75 Мбайт
Системна пам'ять memory	Від 128 Мбайт
Розміри бази даних адрес	Від 8000 MAC-адрес
Вбудована флеш-пам'ять	Від 32 Мбайт
Транків (кількість)	Від 64
VLAN (кількість)	Від 1024
Маршрутизованих VLAN (кількість)	Від 32
Черг (кількість)	Від 8

3.4 Вибір обладнання для побудови комп'ютерної системи

Головне відділення компанії ТОВ «Лелека» розміщено в двоповерховому офісному будинку. Віддалене відділення по роботі з клієнтами компанії розташоване на першому поверсі 9 поверхової будівлі.

Ці будівлі мають підключення до провайдерів Інтернет.

Логічна схема комп'ютерної мережі компанії «Лелека» представлена на рис. 3.1.

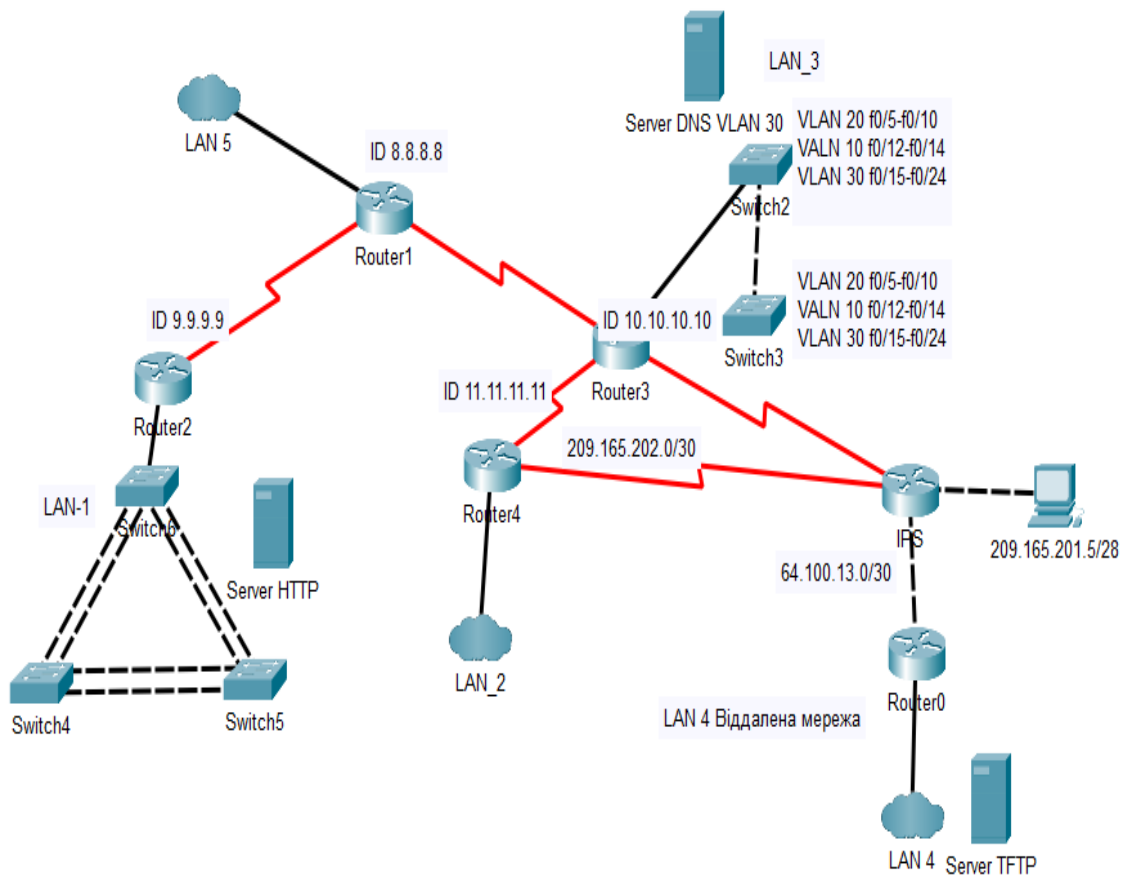


Рисунок 3.1 – Логічна схема комп'ютерної мережі компанії «Лелека»

Для схеми системи було підібрано наступне пасивне і активне мережне обладнання (таблиця 3.3).

Таблиця 3.3 – Пасивне і активне мережне обладнання

Позиція	Найменування і технічна хар-ка	Тип, марка, позначення	Одиниці виміру	Кількість
1	Комутатор	Cisco RV345P Dual WAN Gigabit VPN Router (RV345P-K9-G5)	шт	4
2	Маршрутизатор	D-Link (DES-1210-28P)	шт	2
3	Маршрутизатор	D-Link DGS-1210-52	шт	3
4	Маршрутизатор	D-Link DGS-1008P PoE	шт	1
5	Бездротовий маршрутизатор	D-Link DIR-615	шт	2
6	Сервер	Dell PowerEdge T30 (T30v01)	шт	2
7	Системний блок	Everest Home&Office 1036	шт	90
8	Монітор	QUBE H24F75	шт	90
9	Клавіатура	2E KS 101 USB Black	шт	90
10	Миша	HP X1500 USB Black (H4K66AA)	шт	90
11	Принтер	Canon i-sensys mf641cw	шт	5
12	Гарнітура	Sennheiser pc 5 chat	шт	90
12	ОС робочих станцій	Windows 10 professional	шт	90
13	ОС сервера	Dell windows server 2019 standard rok (634-bsfx)	шт	2
14	ІР-відеокамера	Ір камера Dh-sd1a203t-gn-w	шт	2

В таблиці 3.4 показана схема адресації мережі, а в таблиці 3.5 показана схема адресування підмереж маршрутизаторів.

Таблиця 3.4 – Схема адресації мережі

Назва	Розмір	Вид. розмір	Адреси	Маска	Десяткова маска	Діапазон адрес	Широкомовлення
LAN1	74	126	10.22.194.0	/25	255.255.255.128	10.22.194.1 – 10.22.194.126	10.22.194.127
LAN2	17	30	10.22.194.128	/27	255.255.255.224	10.22.194.129 – 10.22.194.158	10.22.194.159
LAN3	90	126	10.22.193.128	/25	255.255.255.128	10.22.193.129 – 10.22.193.254	10.22.193.255
LAN4	241	254	10.22.192.0	/24	255.255.255.0	10.22.192.1 – 10.22.192.254	10.22.192.255
LAN5	99	126	10.22.193.0	/25	255.255.255.128	10.22.193.1 – 10.22.193.126	10.22.193.127

Таблиця 3.5 – Схема адресування підмереж маршрутизаторів

Назва	Розмір	Виділений розмір	Адреси	Маска	Десяткова маска	Діапазон доступних адрес	Широкомовлення
WAN1	2	2	10.0.8.0	/30	255.255.255.252	10.0.8.1 – 10.0.8.2	10.0.8.3
WAN2	2	2	10.0.8.4	/30	255.255.255.252	10.0.8.5 – 10.0.8.6	10.0.8.7
WAN3	2	2	10.0.8.8	/30	255.255.255.252	10.0.8.9 – 10.0.8.10	10.0.8.11
WAN4	2	2	10.0.8.12	/30	255.255.255.252	10.0.8.13 – 10.0.8.14	10.0.8.15
WAN5	2	2	10.0.8.16	/30	255.255.255.252	10.0.8.17 – 10.0.8.18	10.0.8.19
WAN6	2	2	10.0.8.20	/30	255.255.255.252	10.0.8.21 – 10.0.8.22	10.0.8.23
WAN7	2	2	10.0.8.24	/30	255.255.255.252	10.0.8.25 – 10.0.8.26	10.0.8.27
WAN8	2	2	10.0.8.28	/30	255.255.255.252	10.0.8.29 – 10.0.8.30	10.0.8.31

3.5 Розробка структурної схеми комп'ютерної системи туристичної компанії «Лелека»

Відповідно до вибраного обладнання та спроектованій логічній схемі комп'ютерної мережі (рисунок 3.1) розробимо структурну схему комп'ютерної системи туристичної компанії ТОВ «Лелека» (рисунок 3.2).

IP-камеру підключимо до комутатора LAN3.

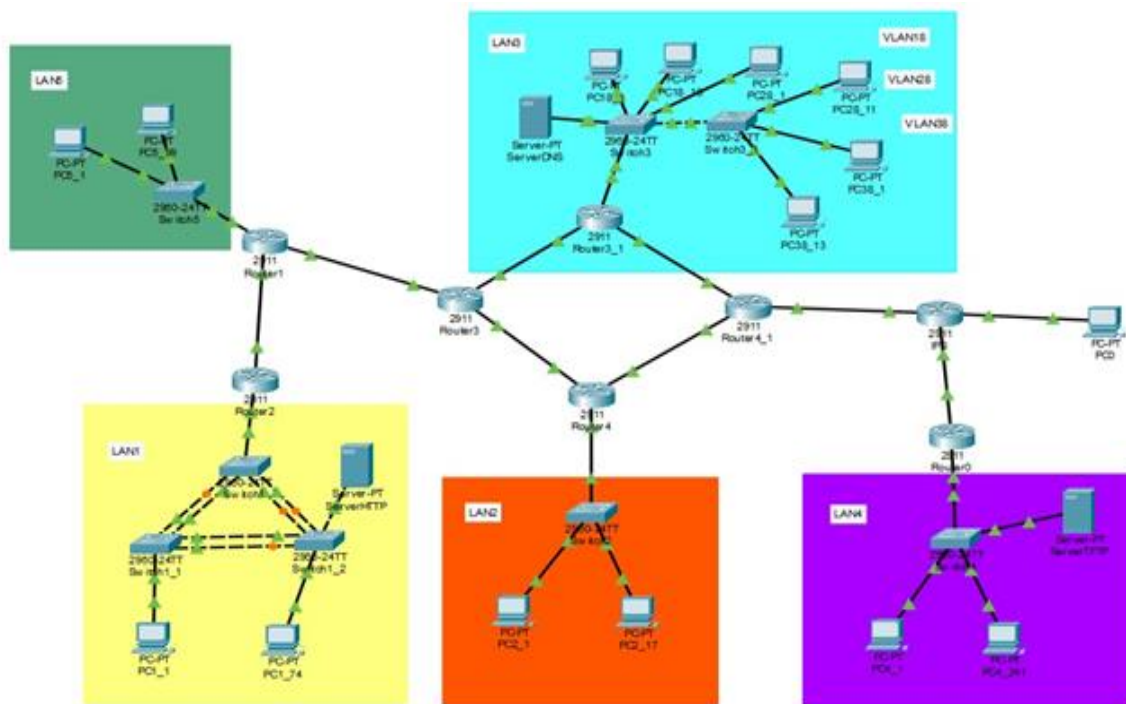


Рисунок 3.2 – Структурна схема комп'ютерної системи компанії «Лелека»

На цій схемі локальна мережа LAN 1 – локальна мережа двоповерхового офісного приміщення туристичної компанії ТОВ «Лелека». LAN 2 - локальна мережа бухгалтерії. LAN 3 – локальна мережа відділу менеджменту. Локальна мережа LAN 4 - об'єднує обладнання, яке встановлене у відділі по роботі з персоналом. Локальна мережа LAN 4 об'єднує головний офіс з віддаленим відділенням по роботі з клієнтами.

4 РАЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Призначення і сфера застосування програми

Програмне забезпечення комп'ютерної системи туристичної компанії «Лелека» розробляється для отримання інформації з відеокамери, виділення фреймов зображень, знаходження них обличь осіб, запису їх в окремий каталог або порівняння їх з обличчями з бази даних і передачі результатів оператору.

4.2 Обґрунтування технічних характеристик програми

4.2.1 Постановка завдання на розробку програми

Для розробки програмного забезпечення комп'ютерної системи компанії «Лелека» була вибрана мова програмування Python з бібліотеками роботи із зображеннями OpenCV, Face recognition, NumPy в середовищі Visual Studio Code.

4.2.2 Структура і алгоритм функціонування програми

Комп'ютерна система повинна працювати повний робочий день і тому структура програми повинна мати циклічний характер.

4.2.3 Вибір складу програмних засобів

Для розробки програми пропонується застосовувати Visual Studio Code або коротко VS Code. Це добрий і популярний редактор коду від компанії Microsoft.

З метою скорочення часу для розробки програми пропонуються застосувати наступні бібліотеки:

- OpenCV;
- Face recognition;
- NumPy.

4.3 Опис розробленої програми

4.3.1 Загальні відомості

Розроблена програма має підпрограми для отримання інформації з відеокамери, виділення фреймов зображень, знаходження них обличчів осіб, запису їх в окремий каталог або порівняння їх з обличчями з бази даних і передачі результатів менеджерам компанії «Лелека».

Текст програми реалізований мовою Python версії 3 в середовищі Visual Studio Code від компанії Microsoft.

4.3.2 Функціональне призначення

4.3.2.1 Типи розв'язуваних завдань

Програма призначена для обробки зображень в туристичній компанії «Лелека», яка застосовувалась під час перевірки комп'ютерної системи та експериментальних досліджень (Додаток А).

4.3.2.2 Функціональні обмеження

Дозволені формати для зображень – jpg, png, а для відеоінформації – mp4.

4.3.3 Опис логічної структури програми ідентифікації осіб

Схема алгоритму роботи програми для ідентифікації осіб на зображенні представлена на рисунку 4.1.

Вона складається з імпортування необхідних бібліотек та основного класу, циклу підготовки, знаходження обличчів на зображенні, їх ідентифікації та збереження результатів.

4.3.4 Використовувані технічні засоби

Для роботи комп'ютерної системи застосовується цифрова камера, яка включена в комп'ютерну мережу туристичної компанії «Лелека».

4.3.5 Цикл роботи програми

Цикл обробки зображень включає: імпортування бібліотек, створення папок для вхідних зображень та вилючених зображень осіб, отримання зображення, читання зображень, виділення обличчя на зображенні, формування навколо нього прямокутної рамки та запис виділеного фрагменту в окремий файл і в окрему папку, якщо програма виявила обличчя (рисунок 4.1) на зображенні.

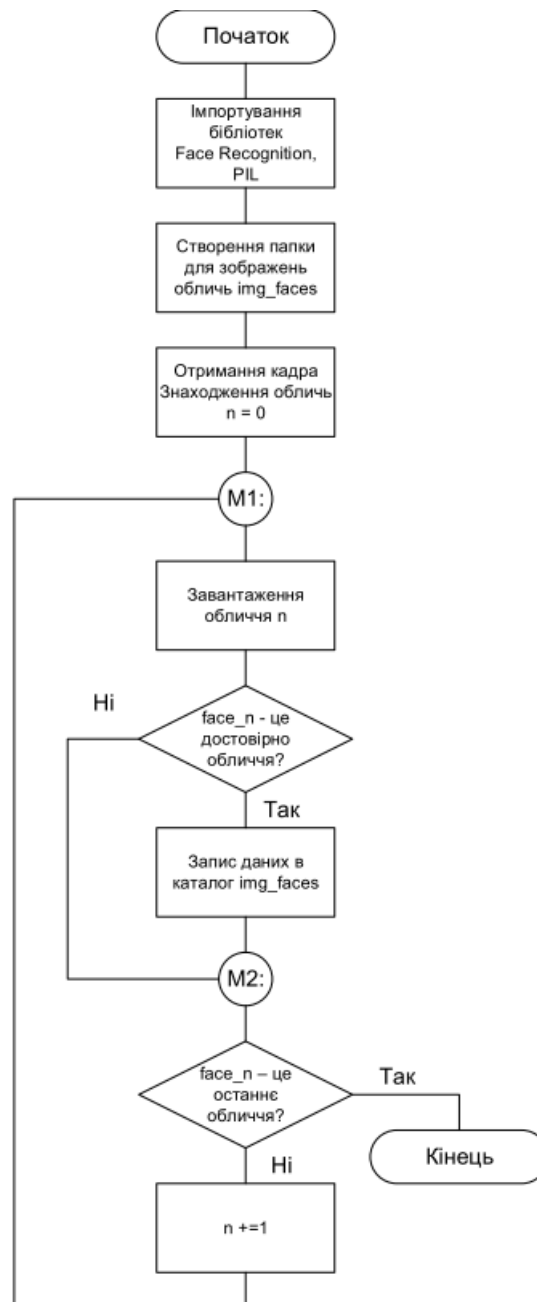


Рисунок 4.1 – Схема алгоритму програми обробки зображень

4.3.6 Вхідні та вихідні дані

Зображення, які отримуються, мають вид масиву $X * Y * Z$ цілих чисел розміром від 0 до 255 пк (пікселей).

При роботі з зображенням, отриманим з відеокамери, у програмі, що розроблена, воно представляється у вигляді масиву X, Y, Z цілих величин від 0 до 255, де X та Y – висота та ширина зображення у пікселях, а Z – колірний компонент, яка дорівнює 3 для формату RGB.

5 ЕКСПЕРИМЕНТАЛЬНИЙ РОЗДІЛ

5.1 Постановка завдання експерименту і обґрунтування методики

Виконаємо перевірку роботи розробленої програми для вилучення зображень осіб та визначимо параметри налаштування комп'ютерної системи. Проведення експерименту необхідне для оцінки розробленої системи та можливості її застосування.

Програму перевіримо як під час роботи з підібраними моделями зображень, так при отриманні реальних відео-файлів.

5.2 Проведення експерименту для перевірки розробленої системи

5.2.1 Перевірка програми по вилученню зображень з потокового відео

Для перевірки розробленої програми застосуємо відео, яке підібрано для експерименту в розділі 2 (рисунок 2.1).

В результаті при завданні дільника $k = 5$, $divinder = 125$, отримаємо 18 кадрів з відео-файлу довжиною 94 с при $fps = 24.92$ кадр./с.

Деякі з них приведено на рисунках (рисунок 5.1 – 5.3).



Рисунок 5.1 – Отриманий кадр №1 з відео-файлу



Рисунок 5.2 – Отриманий кадр №14 з відео-файлу



Рисунок 5.3 – Отриманий кадр №18 з відео-файлу

5.2.2 Перевірка роботи програми по знаходженню обличчів осіб на зображеннях та їх вилученню

Спочатку застосуємо кадри (рисунок 5.1 - 5.3), які вилучені з відео.

Застосуємо розроблену програму і отримаємо зображення з розпізнаними обличчями, які помічені рамками жовтого кольору (рисунок 5.4-5.6).



Рисунок 5.4 – Кадр №1 з розпізнаними обличчями з відео-файлу



Рисунок 5.5 – Кадр №14 з розпізнаними обличчями з відео-файлу



Рисунок 5.6 – Кадр №18 з розпізнаними обличчями з відео-файлу

Зображеннями обличь, які вилучені з кадра №18 і розміщені в окремий каталог, представлені на рисунку 5.7 – для кадра №18 з відео-файлу).



Рисунок 5.7 – Зображеннями обличь з кадра №18, які розміщені в окремий каталог

Далі застосуємо зображення моделі №2, які були підібрані для експерименту в розділі 2 (рисунок 2.2).

Це фотографії осіб з різними положеннями голови (піднята, опущена, повернута вправо, вліво), закритими очами та надітими окулярами та масками.

З використанням розробленого ПО виконаємо пошук обличь моделей №2, №3, №4 та №5 (рисунки 5.8 – 5.11).

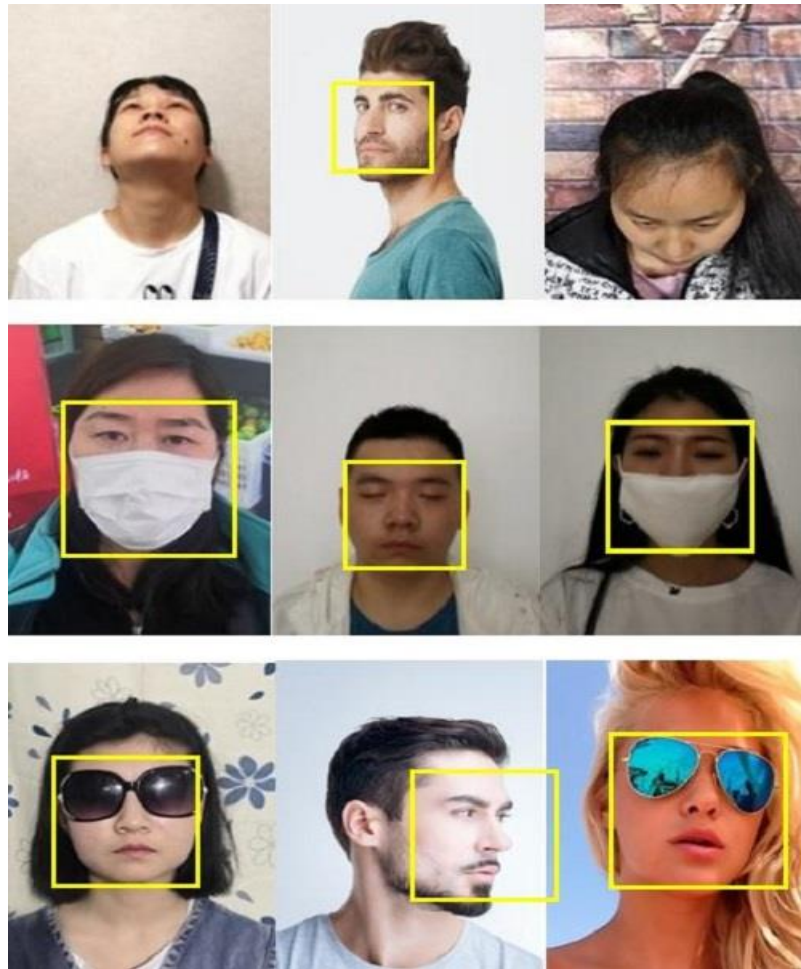


Рисунок 5.8 – Результати розпізнавання обличчя на моделі №2



Рисунок 5.9 – Результати розпізнавання обличчя на моделі №3



Рисунок 5.10 – Результати розпізнавання обличь на моделі №4



Рисунок 5.11 – Результати розпізнавання обличь на моделі №5

5.2.3 Перевірка роботи програми для ідентифікації осіб

Під ідентифікацією розуміється порівняння отриманого зображення обличчя особи яка входить до приміщення туристичної компанії «Лелека» з наявною кількістю зразків моделей обличь співробітників, яким дозволений вхід.

В якості бази пропонується взяти обличчя з моделі №3, а потім №4 (приймаємо що це співробітники компанії), а порівнювати будемо з фотографією ас. Панферової Я.В. (рисунок 5.12), яку взята з сайту кафедри ІТКІ (it.nmu.org.ua). На моделях №3 і №4 ас. Панферова Я.В. присутня.



Рисунок 5.12 – Фотографія для порівняння

В розробленій програмі застосовується бібліотека Face Recognition, робота якої базується на нейромережі dlib.

Зображення кожної особи обробляється навченою згортковою нейронною мережею з отриманням матриці з 128 параметрів. Під час ідентифікації особи порівнюються ці матриці.

В результаті порівняння цієї фотографії (рисунок 5.12) з особами моделі №3 отримаємо наступний результат – рисунок 5.13, 5.14 а для моделі №4 – рисунок 5.15.


```

[-1.31275102e-01  1.03123508e-01  1.09198488e-01 -9.38511044e-02
-1.52321115e-01  1.03970990e-04 -3.57992277e-02 -7.71643668e-02
 1.23011865e-01 -1.01725556e-01  1.23262107e-01 -9.27081630e-02
-2.84969509e-01  3.80815566e-02 -6.27983874e-03  1.94710776e-01
-2.39244044e-01 -1.68166205e-01 -1.58076942e-01 -1.86416432e-02
 3.18121798e-02  1.31654233e-01 -3.49420384e-02  5.29773943e-02
-2.01561317e-01 -3.68366778e-01 -5.35658151e-02 -8.28920379e-02
-9.42573473e-02 -9.94170606e-02  7.03681484e-02  1.05533943e-01
-6.25117496e-02  3.02522629e-02  1.00344226e-01  6.86538294e-02
 1.80151593e-02 -1.03347525e-01  1.61538199e-01 -3.11913062e-03
-2.78842211e-01 -4.02770415e-02  1.50048912e-01  2.67875761e-01
 2.19494343e-01  7.04076001e-03  1.03973383e-02 -8.10867473e-02
 1.05604984e-01 -3.27581376e-01 -6.16078414e-02  1.36335716e-01
 9.02440306e-03  1.30312324e-01  9.37727690e-02 -1.65149987e-01
 5.89791201e-02  9.99080837e-02 -1.30019441e-01 -2.03817412e-02
 8.34929943e-02 -2.13196874e-01  9.92406067e-03 -4.20670398e-02
 1.84735700e-01  1.75769646e-02 -1.32911190e-01 -1.29802018e-01
 1.10629320e-01 -1.72568530e-01 -9.63178426e-02  7.22217262e-02
-1.14925586e-01 -1.15350798e-01 -2.82113582e-01 -8.64865556e-02
 3.94874185e-01  1.31837428e-01 -2.20033288e-01 -2.95364894e-02
-8.73345658e-02  3.32815871e-02  3.83888707e-02  1.49336457e-01
 5.16515225e-04 -8.98932591e-02 -1.04111582e-01  3.62224802e-02
 2.37598836e-01 -6.79898709e-02  8.87977853e-02  2.53664643e-01
 9.28599685e-02 -3.84966694e-02  1.33096445e-02  4.78666313e-02
-1.36288613e-01 -2.52531078e-02 -4.62813005e-02  8.45240988e-03
 2.07112934e-02  1.12168305e-02  3.62261832e-02  1.66258603e-01
-2.02760026e-01  2.01897070e-01 -1.92513056e-02 -4.36180755e-02
 2.99421698e-02 -1.15264067e-03  4.29682657e-02 -7.77725428e-02
 1.58347562e-01 -2.21487477e-01  3.78915109e-02  2.30403900e-01
-4.49919552e-02  1.39707267e-01 -4.57289964e-02  1.20984532e-01
-8.93148929e-02  2.15660334e-02 -1.61945641e-01 -6.63038343e-02
 2.69629192e-02 -1.06595233e-02 -5.51449656e-02 -1.50284134e-02]

```

Рисунок 5.13 – Матриця особи, яка ідентифікується

```

[-0.12517759  0.0761189  0.09924868 -0.07956731 -0.11148012 -0.02962043
-0.02608011 -0.01761599  0.1060145 -0.05944242  0.1783158  0.04494637
-0.33584118 -0.04129354 -0.10160846  0.09550749 -0.04698604 -0.12506761
-0.10022423 -0.07589652  0.03731207  0.03491622 -0.01771244 -0.00931428
-0.11598817 -0.24234192 -0.0901777 -0.06358971 -0.09266893 -0.12867427
 0.01158972  0.07508159 -0.1441987 -0.10120899  0.03370668  0.11983244
-0.05129366  0.01005988  0.24339579  0.07589228 -0.14159502  0.06744482
 0.02599323  0.31522176  0.15336727  0.06112022  0.03005989 -0.02867442
 0.19527179 -0.3999818  0.07427002  0.14054283  0.19684689  0.03048153
 0.11035506 -0.18019618  0.05510417  0.16400771 -0.2349353  0.14454667
 0.01866288 -0.17793897 -0.11291679 -0.02843728  0.16604728  0.11680158
-0.13422942 -0.12057  0.23148857 -0.1278716 -0.03290374  0.17126748
-0.08368027 -0.1951061 -0.23623918  0.15051466  0.39479205  0.19257411
-0.19611622  0.00527599 -0.04729612  0.03789725  0.00659984  0.02542445
-0.1186779  0.02149161 -0.11967287  0.01207616  0.17284478 -0.00673696
-0.07440232  0.25968528  0.08905143  0.13919511  0.07778098  0.03145708
-0.05795423 -0.09108797 -0.17564619  0.01829953  0.11267541 -0.13773461
-0.01166918  0.06173436 -0.11379455  0.11854109 -0.11539998 -0.04112446
-0.11191518 -0.06063283 -0.09462751  0.05054722  0.17538744 -0.28973711
 0.19623697  0.19140075 -0.04649799  0.08842964  0.13938519 -0.03635108
 0.05747793 -0.08081403 -0.13908459 -0.16578808  0.10807882 -0.05257031
 0.06306329  0.00412937]
[False]
Sorry, Sorry, wait for the manager!

```

Рисунок 5.14 – Результат порівняння з першим обличчям

```

2.99421698e-02 -1.15264067e-03 4.29682657e-02 -7.77725428e-02
1.58347562e-01 -2.21487477e-01 3.78915109e-02 2.30403900e-01
-4.49919552e-02 1.39707267e-01 -4.57289964e-02 1.20984532e-01
-8.93148929e-02 2.15660334e-02 -1.61945641e-01 -6.63038343e-02
2.69629192e-02 -1.06595233e-02 -5.51449656e-02 -1.50284134e-02]
[-1.64150015e-01 1.10453829e-01 1.49188966e-01 -8.34174305e-02
-1.60906211e-01 4.40046415e-02 5.38589060e-03 -1.50665373e-01
1.35007739e-01 -1.46021277e-01 1.93518966e-01 2.78755985e-02
-2.57015169e-01 1.17197093e-02 5.40440343e-03 2.03328818e-01
-2.12092504e-01 -1.42783135e-01 -1.46306172e-01 6.81552105e-03
1.00368857e-02 1.67930871e-01 -6.00012578e-02 4.94944938e-02
-1.60571590e-01 -3.20712835e-01 -6.36166409e-02 -5.53854257e-02
-8.34707022e-02 -7.17509240e-02 -4.43917513e-02 3.43970805e-02
-9.03570652e-02 -6.96303323e-03 9.37514082e-02 4.94782701e-02
-2.98474170e-02 -1.20327249e-01 1.54332042e-01 -3.25713353e-03
-2.69829303e-01 -3.45224999e-02 1.57598898e-01 2.01133981e-01
2.52248824e-01 -5.16757257e-02 5.60805202e-05 -1.06055759e-01
1.61965892e-01 -3.47971499e-01 -1.88420545e-02 7.52950311e-02
2.61816457e-02 1.19318061e-01 9.26436707e-02 -2.15454131e-01
-3.87122817e-02 1.22189023e-01 -8.10386539e-02 1.45334424e-02
4.17607985e-02 -2.27187112e-01 -1.46243861e-02 -3.57185863e-02
1.43595099e-01 5.16741425e-02 -1.52671859e-01 -1.44771650e-01
1.36315137e-01 -1.52840316e-01 -1.16194189e-01 2.48447433e-02
-1.02143615e-01 -1.47041544e-01 -2.96685487e-01 -1.14790536e-02
4.37286437e-01 1.24027789e-01 -2.71419972e-01 -3.91258076e-02
-9.00891349e-02 3.54058342e-03 5.51108383e-02 1.76639840e-01
5.95346056e-02 -1.02187395e-01 -9.30389017e-02 -5.34787402e-03
2.63243765e-01 -3.55803482e-02 8.52563456e-02 3.11129749e-01
5.97669333e-02 1.33373775e-03 5.35906777e-02 1.20252810e-01
-1.31892130e-01 -3.76990736e-02 -8.54414105e-02 1.33042140e-02
1.99448317e-04 -2.67612822e-02 2.34344769e-02 2.06793934e-01
-2.02958465e-01 2.24989220e-01 -2.27115154e-02 -8.37346260e-03
5.88573292e-02 -6.20039925e-02 9.65443719e-03 -5.56333847e-02
1.29181847e-01 -1.97463483e-01 9.51240435e-02 1.80695206e-01
-7.04266354e-02 1.80215523e-01 -1.11200586e-02 1.06261320e-01
-3.88737619e-02 1.73686557e-02 -1.78043872e-01 -9.22864899e-02
6.26081452e-02 -5.53833209e-02 -4.38773483e-02 8.60313419e-04]
[True]
Welcome to the Travel company Leleka!

```

Рисунок 5.14 – Результат порівняння з третім обличчям

5.3 Оцінка результатів

5.3.1 Вилучення зображень з потокового відео

Стосовно вилученню зображень з потокового відео можна зробити наступні розрахунки. Враховуючи отримані дані визначимо значення множника k , значення якого показує через який час потрібно з камери отримувати наступний кадр.

Будемо рахувати, що після входу відвідувача або співробітника в приміщення компанії «Лелека» з відеокамери необхідно отримати не менше 5 – ти кадрів. Відстань від дверей до турнікету чергового на прохідній складає 3 м.

Середня швидкість пішохода по рівній поверхні становить 90 – 120 кроків за хв. або 4 – 5.5 км/год.

Прийmemo, що вона дорівнює 5.5 км/год. = 1.5 м/с. Тобто людина пройде цю відстань десь за 2 с. і з відеокамери необхідно отримати 5 кадрів.

Тоді, якщо

$$n = \text{frame_id} / (\text{fps} * k),$$

де n – номер зображення, яке потрібно отримати;

frame_id – номер поточного кадру;

fps – швидкість передачі кадрів, кадр./с;

k – множник,

то для заданих умов на рівні турнікету для $n = 5$:

$$k = \text{frame_id} / \text{fps} * n = 50 / 25 * 5 = 0.4.$$

А множник треба внести в програму, що б отримувати в умовах компанії «Лелека» роботи комп'ютерної системи необхідну кількість кадрів для ідентифікації осіб, які проходять через турнікет чергового.

5.3.2 Знаходженню обличь на зображеннях

При виконанні експериментів було виконано за допомогою розробленої програми знаходження обличь на різних зображеннях. Проаналізуємо якість знаходження обличь для моделей де кількість осіб було більше 10 і занесемо результати в таблицю 5.1.

Для аналізу застосуємо результати експериментів з моделями №2, №3, №4, №5.

Таблиця 5.1 – Результати експериментів по знаходженню обличь

№ моделі	Рисунок з результатом знаходження	Формат файлу	Розмір зображення, пкс	Кількість осіб на зображенні	Знайдено обличь	
					осіб	%
3	5.9	jpg	3041 x 1889	14	14	100
4	5.10	jpg	2048 x 1365	53	50	94.3
5	5.11	jpg	1638 x 1092	78	62	79.5
						91.3

Аналіз результатів показує, що при отриманні зображень для знаходження на них обличь треба враховувати відстань на якій встановлена камера.

При встановлені відеокамери на відстані 3-5 м, як це буде в умовах компанії «Лелека», дає практично 100% результат знаходження обличчя, див. модель №3, рисунок 5.9.

Зниження показників достовірності виявлення обличь до 94.3% можливо за рахунок отримання зображень з частково закритим обличчям, див. модель №4, рисунок 5.10.

Але цього не буде в умовах компанії «Лелека» під час проходження через турнікет чергового на прохідній, так як там фізично складаються умови по відкриттю обличчя кожного відвідувача або співробітника.

Крім того виключаються варіанти з високо піднятим або низько опущеним обличчям, як це було в моделі №2, рисунок 5.8.

Зниження показників знаходження обличчя до 79.5 для моделі №5 (рисунок 5.11) обумовлено далекою відстанню до відеокамери що зменшило розміри обличч осіб, які знаходилися в останньому ряду (в компанії «Лелека» це виключається) і зниженою освітленістю. Вимогу про достатнє освітлення закладено в проект.

Виконання цих умов забезпечить ідентифікацію осіб з достовірністю виявлення обличч не нижче 99.38% розробленою програмою із застосуванням бібліотеки Face Recognition на базі нейромережі dlib.

ВИСНОВКИ

Атестаційна робота магістра є завершеною роботою в якій вирішена задача – обґрунтування параметрів комп'ютерної системи туристичної компанії «Лелека».

Основні висновки і результати роботи полягають в наступному:

1. Проведено аналіз галузі та умов застосування системи та огляд об'єкта впровадження.
2. Виконано аналіз особливостей запису кадрів зображення з потокового відео камери;
3. Вивчено особливості знаходження обличчя на вилучених зображеннях мово Python, вилучення та запис їх в окремий каталог.
4. Розглянуто процес порівняння записаних обличчя з базами наявних співробітників компанії або небезпечних осіб.
5. Розроблена структура комп'ютерної системи компанії «Лелека».
6. Розроблено комплекс програм підсистеми відеоконтролю.
7. Проведено експериментальні дослідження і обґрунтуванні параметри комп'ютерної системи туристичної компанії «Лелека».
8. Розроблена комп'ютерна система забезпечує ідентифікацію осіб з достовірністю виявлення обличчя не нижче 99.38% розробленою програмою із застосуванням бібліотеки Face Recognition на базі нейромережі dlib.

ПЕРЕЛІК ПОСИЛАНЬ

1. Журавська І. М. Проектування та монтаж локальних комп'ютерних мереж :[навчальний посібник] / І. М. Журавська. – Миколаїв : Видавництво ЧДУ ім. Петра Могили, 2016. – 396 с.
2. Martin Casado, Tal Garfinkel, Aditya Akella, Michael J. Freedman Dan Boneh, Nick McKeown, Scott Shenker. SANE: A Protection Architecture for Enterprise Networks, 15 Usenix Security Symposium, Vancouver, Canada, August 2006.
3. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014-2019 [Електронний ресурс] // Cisco Systems, Inc [Офіц. сайт]. URL: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/vism-networking-index-vni/white_paper_c11-520862.html
4. Guangcan Mai, Kai Cao, Pong C Yuen, and Anil K Jain. On the reconstruction of face images from deep face templates. *TPAMI*, 41(5):1188–1202, 2018.
5. O. M. Parkhi, A. Vedaldi, A. Zisserman Deep Face Recognition British Machine Vision Conference, 2015.
6. Жуков, І. А. Комп'ютерні мережі та технології : навч. посіб. / І.А. Жуков, В. О. Гуменюк, І. Є. Альтман. – К. : НАУ, 2004. – 276 с.
7. Мережеве обладнання [Електронний ресурс] – Режим доступу: URL: https://elmir.ua/routers/router_zyxel_sbg5500-a.html. – Загол. з екрану.
8. Цвіркун Л.І. Комп'ютерні мережі. Методичні рекомендації до виконання курсового проекту студентами галузі знань 12 Інформаційні технології спеціальності 123 Комп'ютерна інженерія / Л.І. Цвіркун, Я.В. Панферова, Л.В. Бешта ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро: НТУ «ДП», 2018. – 28 с.
9. Цвіркун Л.І. Комп'ютерні мережі. Методичні рекомендації до виконання лабораторних робіт студентами галузі знань 12 Інформаційні технології спеціальності 123 Комп'ютерна інженерія: у 2 ч. / Л.І. Цвіркун,

Я.В. Панферова ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро: НТУ «ДП», 2018. – Ч. 1. – 60 с.

10. Атестація здобувачів вищої освіти. Методичні рекомендації до виконання кваліфікаційної роботи бакалавра студентами галузі знань 12 інформаційні технології спеціальності 123 комп'ютерна інженерія / Л.І. Цвіркун, С.М. Ткаченко, Я.В. Панферова, Д.О. Бешта, Л.В. Бешта. – Д.: НТУ «ДП», 2022. – 62 с.

11. Цвіркун Л.І. Інженерна та комп'ютерна графіка. AutoCAD : навч. посіб. / Л.І. Цвіркун, Л.В. Бешта ; під заг. ред. Л.І. Цвіркуна ; М-во освіти і науки України, НТУ «Дніпровська політехніка». – Дніпро: НТУ «ДП», 2018. – 209 с. – ISBN 978-966-350-663-0.

12. Цвіркун Л.І. Розробка програмного забезпечення комп'ютерних систем. Програмування: навч. посіб. [Електронний ресурс] / Л.І. Цвіркун, А.А. Євстігнєєва, Я.В. Панферова ; під заг. ред. проф. Л.І. Цвіркуна ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – 1 електрон. опт. диск (CD-ROM) ; 12 см. – Систем. вимоги (мінімальні): Процесор 32-розрядний (x86) 233 МГц ; 512 МБ RAM ; 128 МБ Video ; від 4-х до 48-х CD-ROM ; Windows 7. – Назва з контейнера. – Дніпро: НТУ «ДП», 2019. – ISBN 978-966-350-638-8.

Додаток А

Текст програми

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ДНІПРОВСЬКА ПОЛІТЕХНІКА»

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ
НАЛАШТУВАННЯ МЕРЕЖІ КОМП'ЮТЕРНОЇ СИСТЕМИ

Текст програми

804.02070743.23011-01 12 01

Листів 8

2023

АНОТАЦІЯ

Програма містить в собі частину програмного коду для вилучення зображень осіб з потокового відео з відеокамери, визначення наявності обличь на зображенні і їх вилучення та порівняння обличь двох осіб або ідентифікації в комп'ютерній системі туристичної компанії «Лелека».

Застосована мова програмування Python 3.11 з бібліотеками роботи із зображеннями OpenCV, Face recognition, NumPy в середовищі Visual Studio Code.

ЗМІСТ

	Стор.
1. Модуль take_img_from_video	4
2. Модуль face_rec	6
3. Модуль extracting_faces	8
4. Модуль compare_faces	9

#take_img_from_video

```
#Завантаження бібліотек
```

```
import os
```

```
import pickle
```

```
import sys
```

```
import face_recognition
```

```
import cv2 as cv
```

```
import numpy as np
```

```
from pathlib import Path
```

```
def take_img_from_video(video_path):
```

```
    #Завантаження відео
```

```
    cap = cv.VideoCapture(video_path)
```

```
    wave = Path(video_path)
```

```
    count = 1
```

```
    #Перевірка та створення каталогу video_img
```

```
    if not os.path.exists("video_img"):
```

```
        os.mkdir("video_img")
```

```
    while (cap.isOpened()):
```

```
        ret, frame = cap.read()
```

```
        fps = cap.get(cv.CAP_PROP_FPS)
```

```
        divider = int((round(fps)) * 5)
```

```
    # Цикл із запису кадрів
```

```

if ret:
    frame_id = int(round(cap.get(1)))
    cv.imshow("Frame", frame)
    print(f"fps = {fps}, divider = {divider}, frame_id = {frame_id}")
    k = cv.waitKey(4)

    if frame_id % divider == 0:
        cv.imwrite(f"video_img/{count}_{wave.name}.jpg", frame)
        print(f"Take a screenshot {count}_{wave.name}")
        count += 1

    if k == ord(""):
        cv.imwrite(f"video_img/{count}_br_{wave.name}.jpg", frame)
        print(f"Take a screenshot br_{count}_{wave.name}")

        count += 1

    if k == ord("q"):
        print(f"Q pressed, end of the app")
        break

else:
    print("It's end!")
    break

cap.release()
cv.destroyAllWindows()

def main():

```

```
take_img_from_video("video/video.mp4")

if __name__ == '__main__':
    main()

# face_rec

#Завантаження бібліотек
import os
import face_recognition
from PIL import Image, ImageDraw

def face_rec():
    file1 = "img/model3.jpg"
    file1_n = "img_upd/model3_n.jpg"

    file2 = "img/model4.jpg"
    file2_n = "img_upd/model4_n.jpg"

#Перевірка та створення каталогу
    if not os.path.exists("img"):
        print("[ERROR] there is no directory 'img'")

    if not os.path.exists("img_upd"):
        print("New directory img_upd created!")
        os.makedirs('img_upd')

    img1 = face_recognition.load_image_file(file1)
```

```
img1_location = face_recognition.face_locations(img1)

img2 = face_recognition.load_image_file(file2)
img2_location = face_recognition.face_locations(img2)

print(img1_location)
print(img2_location)
print(f"Found {len(img1_location)} face(s) in this image")
print(f"Found {len(img2_location)} face(s) in this image")

pil_img1 = Image.fromarray(img1)
draw1 = ImageDraw.Draw(pil_img1)

for(top, right, bottom, left) in img1_location:
    draw1.rectangle(((left, top), (right, bottom)), outline=(255, 255, 0),
width=3)

del draw1
pil_img1.save(file1_n)

pil_img2 = Image.fromarray(img2)
draw2 = ImageDraw.Draw(pil_img2)

for(top, right, bottom, left) in img2_location:
    draw2.rectangle(((left, top), (right, bottom)), outline=(255, 255, 0),
width=3)

del draw2
pil_img2.save(file2_n)
```

```
def main():  
    face_rec()  
  
if __name__ == '__main__':  
    main()
```

#extracting_faces

#Завантаження бібліотек

```
import os  
from pathlib import Path  
import face_recognition  
from PIL import Image, ImageDraw  
def extracting_faces(img_path):
```

#Перевірка та створення каталогу

```
    if not os.path.exists("img"):  
        print("[ERROR] there is no directory 'img'")
```

```
    if not os.path.exists("img_faces"):  
        print("New directory img_faces created!")  
        os.makedirs('img_faces')
```

```
    count = 0
```

```
    faces = face_recognition.load_image_file(img_path)
```

```
faces_locations = face_recognition.face_locations(faces)
```

```
    # Цикл вилучення обличь
```

```
for face_location in faces_locations:
```

```
    top, right, bottom, left = face_location
```

```
    face_img = faces[top:bottom, left:right]
```

```
    pil_img = Image.fromarray(face_img)
```

```
    pil_img.save(f"img_faces/{count}_{wave.name}")
```

```
    count += 1
```

```
    print(count)
```

```
print (f"Found {count} face(s) in this photo")
```

```
def main():
```

```
    extracting_faces("img/18_video.mp4.jpg")
```

```
if __name__ == '__main__':
```

```
    main()
```

#compare_faces

```
#Завантаження бібліотек
```

```
import os
```

```
import face_recognition
```

```
from PIL import Image, ImageDraw
```

```
def compare_faces(img1_path, img2_path):

    img1 = face_recognition.load_image_file(img1_path)
    img1_encodings = face_recognition.face_encodings(img1)[0]
    print(img1_encodings)

    img2 = face_recognition.load_image_file(img2_path)
    img2_encodings = face_recognition.face_encodings(img2)[2]
    print(img2_encodings)

    result = face_recognition.compare_faces([img1_encodings],
img2_encodings)
    print(result)

    if result[0]:
        print("Welcome to the Travel company Leleka!")
    else:
        print("Sorry, wait for the manager!")

def main():
    compare_faces("img/0_jna.jpg", "img/model3.jpg")

if __name__ == '__main__':
    main()
```