МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет прикладної математики та інформатики
(повне найменування назва факультету)

Кафедра інформаційних систем
(повна назва кафедри)

Магістерська робота

«Застосування машинного навчання для великих масивів зображень»

Виконав: студент групи ПМІм - 22
спеціальності
122 – «Комп'ютерні науки»
(шифр і назва спеціальності)

Горбан Н. М.
(підпис) (прізвище та ініціали)

Керівник Венгерський П. С.
(підпис) (прізвище та ініціали)

Рецензент Пелешко Д.Д.
(підпис) (прізвище та ініціали)

ДЕКАН

Львів – 2022

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет прикладної математики та інформатики
Кафедра інформаційних систем
Спеціальність 122 – «Комп'ютерні науки»
(шифр і назва)

«ЗАТВЕРДЖУЮ»
Завідувач кафедри проф. Шинкаренко Г.А
"05" 09 **2022 року**

З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Горбану Назарію Мироновичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Застосування машинного навчання для великих масивів зображень*

керівник роботи *Венгерський Петро Сергійович, доктор фіз-мат. наук, професор*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені Вченою радою факультету від " _____ " _____ 20__ року №_____

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи *дослідження та аналіз сучасних систем, які б дозволяли вибирати деяку підмножину зображень із великої їх кількості, а також аналіз перспектив та можливостей створення власної такої системи, яка б не вимагала значних обчислювальних ресурсів.*

4. Зміст магістерської роботи (перелік питань, які потрібно розробити)

а) Вступ;
б) Фільтри Габора;
в) Обробка зображень;
г) Машинне навчання;
ґ) Оптимізація навчання;
д) Висновки;

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

а) Приклади тестових зображень;
б) Приклади оброблених зображень;
в) Зображення фільтрів;
г) Зображення матриць невідповідності;
ґ) Зображення оптимізаційних діаграм;
д) Таблиця результатів;

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання *15 вересня 2022 року*

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1.	Аналіз літературних джерел за темою магістерської роботи	*вересень*	*Виконано*
2.	Дослідження фільтрів Ґабора	*вересень*	*Виконано*
3.	Аналіз алгоритмів машинного навчання	*жовтень*	*Виконано*
4.	Тестування алгоритмів машинного навчання	*жовтень*	*Виконано*
5.	Пошук можливостей для оптимізації навчання	*жовтень*	*Виконано*
6.	Реалізація оптимізаційного циклу	*листопад*	*Виконано*
7.	Аналіз результатів оптимізації	*листопад*	*Виконано*
8.	Написання та оформлення тексту магістерської роботи	*листопад-грудень*	*Виконано*

Студент _____ Горбан Н. М.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Венгерський П.С

ЗМІСТ

Вступ

Ні для кого не секрет, що у наш час дуже стрімко розвиваються інформаційні технології. Як наслідок цього, щодня кількість різноманітного мультимедійного контенту росте неймовірними темпами. Особливу увагу варто звернути на швидкість темпів росту об’ємів сховищ із зображеннями різного характеру, цей ріст пов’язаний з тим, що такий тип даних другий після тексту, за простотою копіювання та розповсюдження. Ці банки картинок можуть містити найрізноманітніший контент починаючи зі світлин флори та фауни, зроблених професійними фотографами, і закінчуючи фотографіями на медичну тематику, наприклад, із захворюваннями сітківки ока, меланомами тощо .

На жаль переважна більшість усіх цих зображень не містить жодних тегів чи будь-якої іншої анотації, яка б дозволяла з’ясувати вміст зображення не переглядаючи його. А у випадках, коли зображення все-таки мають якийсь опис, його або все одно не достатньо, або він може бути не правильно інтерпретований користувачем. Тому у будь-якому випадку користувачу доведеться переглядати усі зображення самостійно. І якщо виникне потреба з цього великого набору картинок комусь вибирати фотографії для виставки у галереї чи показу прикладів студентам-медикам, то цій людині не стане ні часу, ні фізичних ресурсів, щоб це зробити.

Великі корпорації, такі як Google, для схожих задач класифікації використовують громіздкі згорткові нейронні мережі, які навчаються на величезних наборах даних. На додачу, таке навчання вимагає великих обчислювальних ресурсів. Як можна зрозуміти середньостатистичний користувач не володіє ні такою великою кількістю зображень для тренування мережі, ні достатньою кількістю обчислювальних ресурсів для використання таких моделей.

З огляду на все вищесказане проблема створення системи, яка була б відносно невеликою і при цьому дозволяла б з хорошою точністю класифікувати зображення є досить актуальною. Така система може значно полегшити пошук

зображень, покращити відповідність, знайдених зображень, критеріям пошуку та зекономити найважливіший людський ресурс — час.

Завдання магістерської роботи

Метою цієї роботи була поставлена розробка інформаційної системи, яка б дозволяла класифікувати та вибирати серед великих масивів зображень ті, які б підходили під задані критерії.

Критеріями можуть виступати: наявність на зображенні рослин, тварин, споруд, а також більш конкретні їх ознаки, такі як: різновиди для фауни та флори чи наявність певних елементів (шпилів, арок тощо) для архітектури. У загальному для тренування критеріями виступають спільні ознаки отримані з наперед заданих зображень.

1 Фільтри Ґабора

Фільтр Ґабора – це лінійний фільтр, який використовується для обробки зображень, а саме для виявлення країв, аналізу текстури та виділення ознак. Основною особливістю даного фільтра є те, що він фактично імітує процеси, які відбуваються у корі головного мозку під час того, як ми розпізнаємо лінії та різноманітні текстури довкола нас[4]. Двовимірний фільтр Ґабора можна представити за допомогою наступної формули:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = e^{-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}} \cos(2\pi \frac{x'}{\lambda} + \psi) \tag{1}$$

де

$$x' = x\cos\theta + y\sin\theta \tag{2}$$

і

$$y' = -x\sin\theta + y\cos\theta \tag{3}$$

Розглянемо за що відповідають параметри цієї функції і як їхня зміна впливає на вигляд фільтра. А також дослідимо, яким чином зміна фільтра впливає на зображення, на яке він накладається. Для цього використаємо зображення зебри, оскільки воно насичене різними текстурами та лініями.

Рисунок 1.1 - Тестове зображення

λ - довжина хвилі, визначає ширину смуг функції Ґабора. Збільшення довжини хвилі утворює товстіші смуги, а зменшення – тонші. Розглянемо як змінюється вигляд фільтра, при зміні параметра лямбда на 30, 60 і 100.

Рисунок 1.2 - Вигляд фільтра при значеннях параметра лямбда 30, 60 та 100

При накладанні цих фільтрів, можна помітити, що зі зміною значення параметра лямбда видобуті ознаки стають менш чіткими та розмитими на краях, разом з тим, видобувається менше ознак із заднього фону.

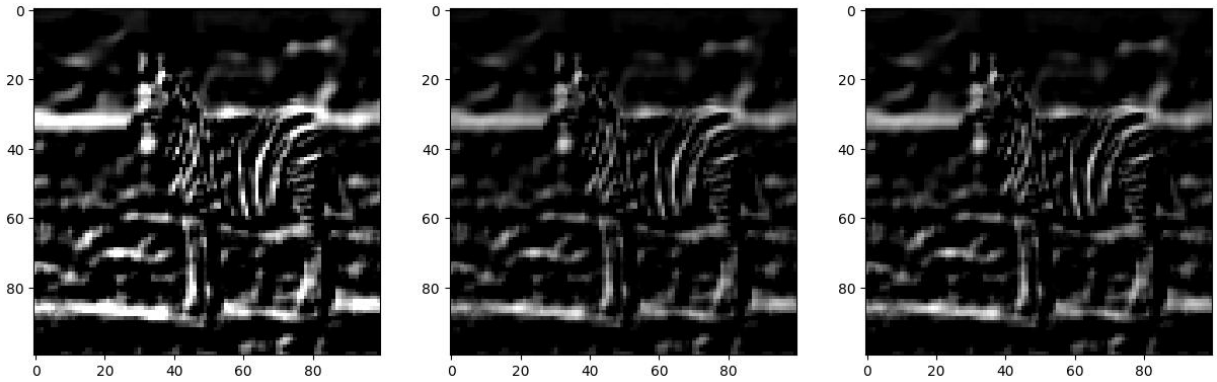

Рисунок 1.3 -Вплив фільтрів з різними значеннями лямбди на зображення

Θ - тета контролює орієнтацію функції Ґабора. На наступному зображенні показано, як змінюється вигляд фільтра при значеннях параметра тета 0, 45, 90.

Рисунок 1.4 - Вигляд фільтра при значеннях параметра тета 0, 45 та 90

Зі зміною даного параметра змінюється характер ліній видобутих із зображення.

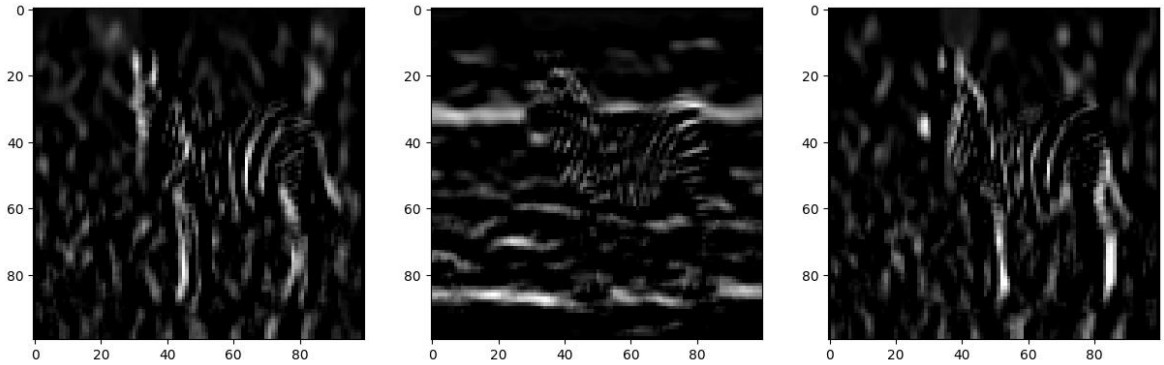

Рисунок 1.5 - Вплив фільтрів з різними значеннями тети на зображення

γ - співвідношення сторін. Цей параметр контролює висоту функції Ґабора. Чим більший параметр гамма, тим меншою буде висота фільтра. При зміні значення цього параметра на 0.25, 0.5 і 0.75 висота функції Ґабора зменшується.

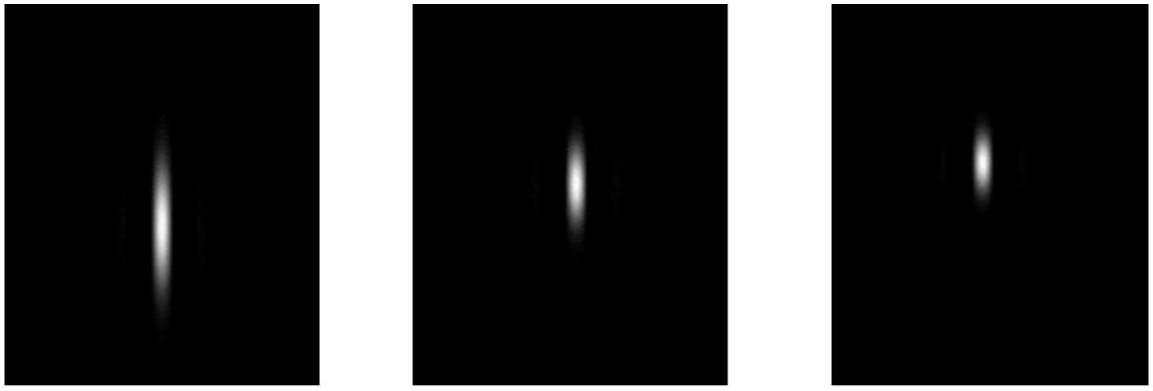

Рисунок 1.6 - Вигляд фільтра при значеннях параметра гамма 0.25, 0.5, 0.75

Збільшення параметра гамма призводить до того, що видобуті ознаки стають чіткішими, однак разом з тим у наборі ознак з’являються шуми.

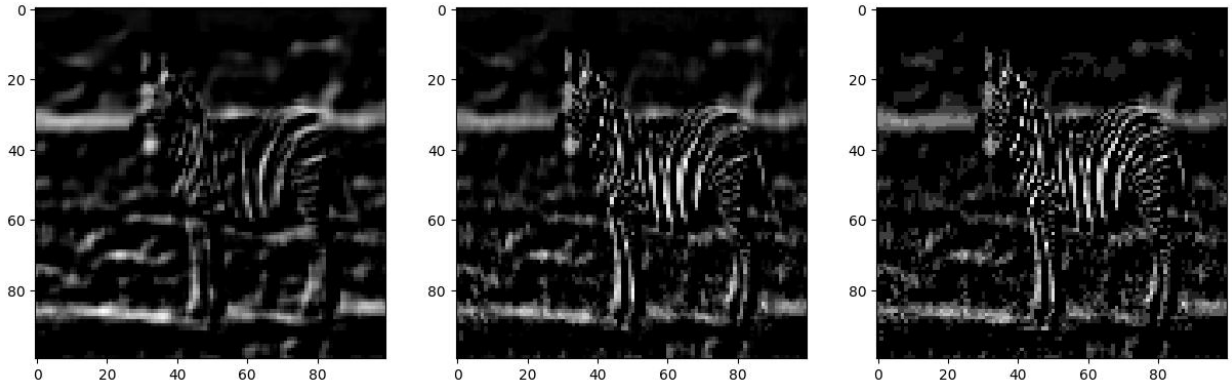

Рисунок 1.7 - Вплив фільтрів з різними значеннями гамми на зображення

σ - пропускна здатність. Даний параметр контролює кількість смуг на фільтрі. При збільшенні сигми від 10 до 30 і 45 кількість смуг у функції Ґабора збільшується.

Рисунок 1.8 - Вигляд фільтра при значеннях параметра сигма 10, 30 та 45

На рисунку 1.9 можна побачити, що зі збільшенням значення параметра сигма видобуті ознаки стають чіткішими, але водночас збільшується і кількість добутих нерелевантних ознак із заднього фону.

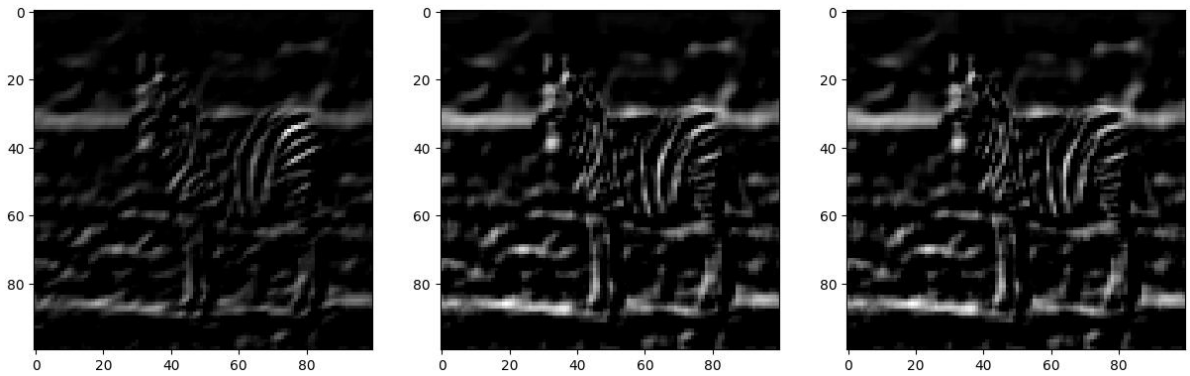

Рисунок 1.9 - Вплив фільтрів з різними значеннями сигми на зображення

Ψ — зсув фази синусоїдальної функції. При зміні значення даного параметра на 0, 90 та 180 градусів фільтр змінюється наступним чином.

Рисунок 1.10 - Вигляд фільтра при значеннях параметра псі 0, 90 та 180 градусів

При значенні параметра псі 0 усе зображення добулось як одна текстура і в результаті було отримано майже повністю біле зображення, а при значенні - 180 із зображення не добулось взагалі нічого. Значення псі 90 у цьому випадку виявилось найкращим, щоб можна було отримати обриси зебри із зображення.

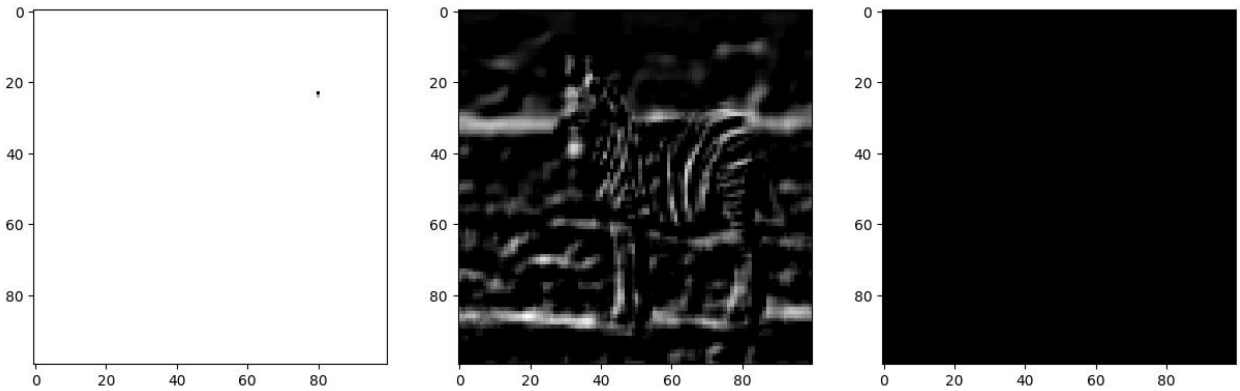

Рисунок 1.11 - Вплив фільтрів з різними значеннями псі на зображення

Щоб краще зрозуміти як використання фільтрів із різними комбінаціями параметрів впливає на зображення, розглянемо наступну картинку.

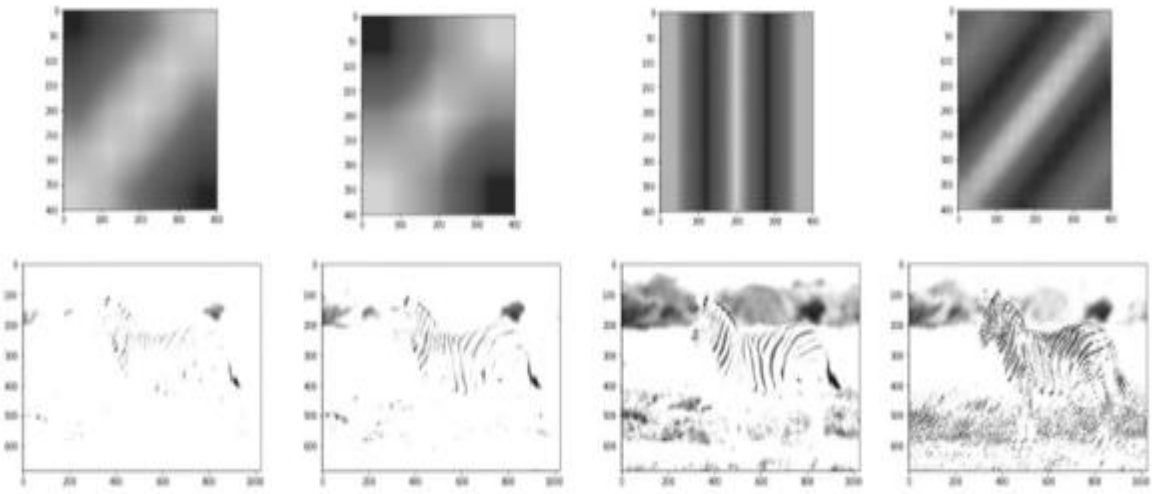

Рисунок 1.12 - Приклад впливів деяких фільтрів на зображення

Можна побачити, що при використанні різних фільтрів із вхідного фото видобуваються зовсім різні ознаки. Зрозуміло, що можливо підібрати таку комбінацію параметрів, щоб видобувати із зображення саме ті ознаки, які потрібно дослідити, або близькі до них. Так, досить добре були видобуті обриси зебри на рисунку 1.12.

З вищенаведених прикладів зрозуміло, що особливо помітним є вплив параметра тета, оскільки він впливає на характер та орієнтацію видобутих ліній із вхідного зображення. Крім того, також досить сильно на зображення впливає значення пропускної здатності, сигма. На рисунку. 1.12, чітко видно, що зі збільшенням пропускної здатності, ознаки добуті із зображення стають чіткішими, однак, разом з тим збільшується і кількість отриманої нерелевантної інформації, такої як, наприклад, вигляд навколишнього оточення, що у результаті може призвести до не правильної класифікації зображень.

2 Обробка зображень

Попередня підготовка зображень для навчання відбувалась у декілька нескладних кроків.

Насамперед генерувався набір із 72 фільтрів розміру 5 на 5 пікселі з різними параметрами. Велика кількість фільтрів дозволяє видобути якомога більше різних ознак, що в свою чергу дозволяє дістати із зображень релевантні ознаки, які сприятимуть правильній класифікації зображень.

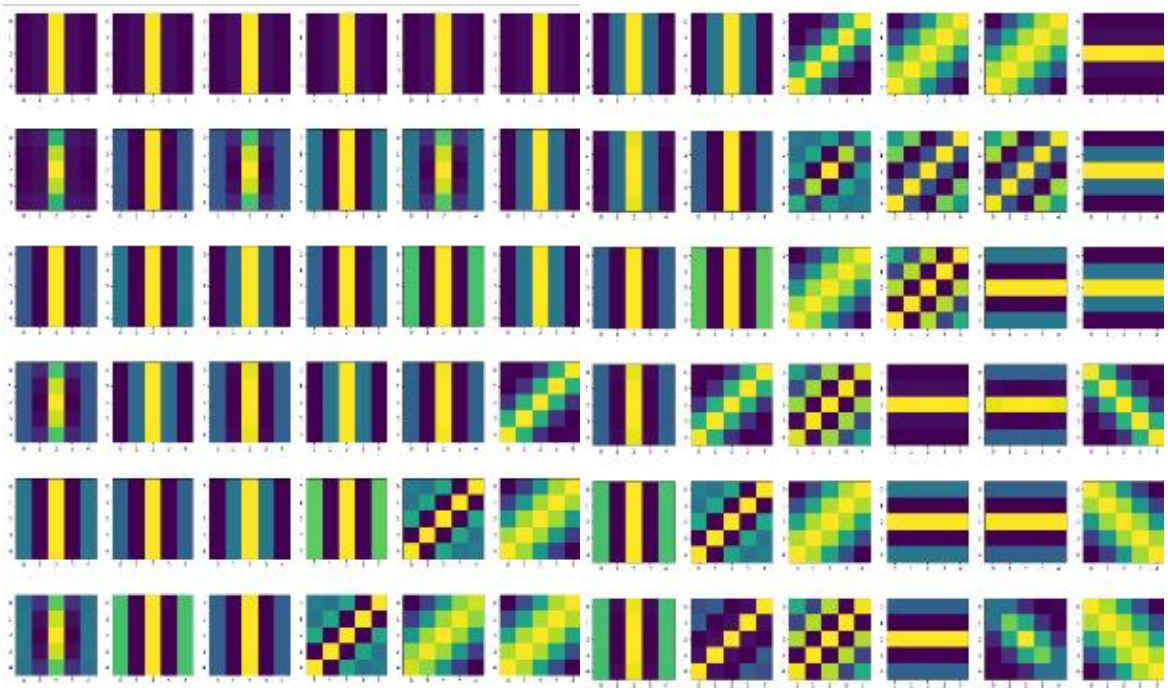

Рисунок 2.1- Вигляд згенерованих фільтрів

Далі, перед тим, як використовувати зображення, його необхідно було привести до деякого стандартизованого формату. Така стандартизація проходила у три прості кроки. Насамперед вхідне зображення перетворювалось у чорно-біле, цей крок допоміг зменшити кількість операцій, як під час подальшої обробки, так і при накладанні фільтрів та навчанні моделі, оскільки тепер замість трьох каналів буде використовуватись лише один. Далі змінювався розмір зображення, це знову ж таки полегшувало навчання моделі. А також цей крок є необхідним, оскільки для машинного навчання вхідні дані повинні бути деякого наперед заданого розміру. У даній роботі я обрав за стандарт розмір зображення 32 на 32 пікселі. І

насамкінець, для того, щоб уникнути випадків, коли зображення або деякі його ділянки є надто темними або надто світлими я проводив його нормалізацію. Після усіх цих перетворень, вхідне зображення змінюється так, як показано на рисунку 2.2.

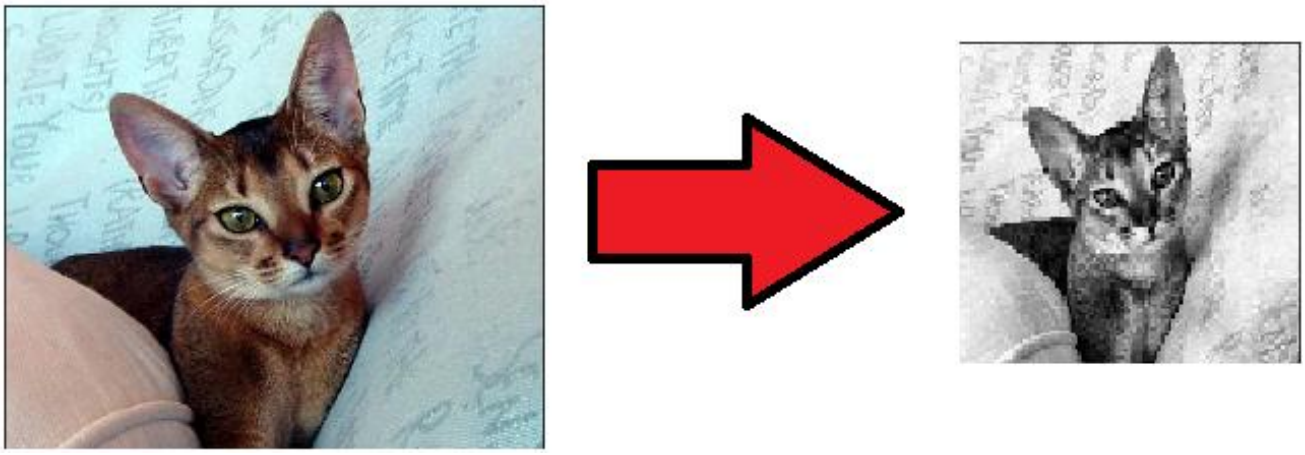

Рисунок 2.2 - Приклад обробки зображення

В результаті проведення усіх цих дій над зображенням, воно стає готовим до того, щоб на нього можна було накладати раніше згенеровані фільтри, таким чином отримується набір ознак видобутих із картинки. Приклад отриманих ознак можна побачити на рисунку 2.3.

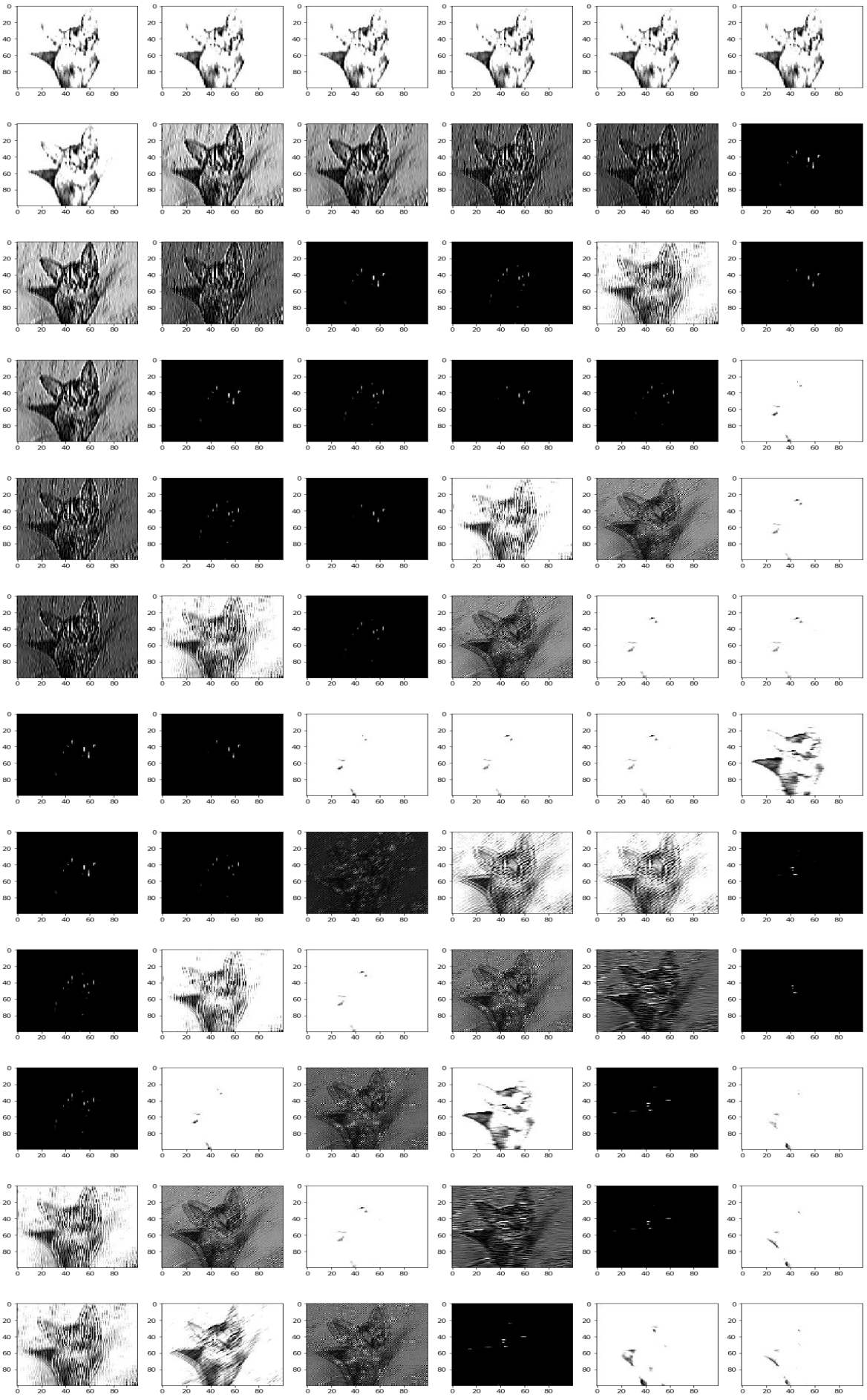

Рисунок 2.3 - Вигляд набору ознак для одного зображення

Насамкінець, отримані набори ознак розкладаються в одновимірні масиви.

Original Image	Gabor 0.0 1 0.7853981633974483 0.05	Gabor 0.0 1 0.7853981633974483 0.5	Gabor 0.0 1 1.5707963267948966 0.05	Gabor 0.0 1 1.5707963267948966 0.5	Gabor 0.0 1 2.356194490192345 0.05	Gabor 0.0 1 2.356194490192345 0.5	Gabor 0.0 3 1.5707963267948966 0.05	...	Gabor 2.356194490192345 3 1.5707963267948966 0.05	Gabor 2.356194490192345 3 1.5707963267948966 0.5	Gabor 2.356194490192345 3 2.356194490192345 0.05	Gabor 2.356194490192345 3 2.356194490192345 0.5	Gabor 2.356194490192345 5 0.7853981633974483 0.05	Gabor 2.356194490192345 5 0.7853981633974483 0.5
39	224	168	165	112	162	102	34	...	184	165	74	59	255	255
66	70	63	0	0	0	0	0	...	0	0	95	93	255	255
30	129	78	41	0	30	0	0	...	168	174	79	66	255	255
71	183	157	89	56	77	54	0	...	0	0	95	91	255	255
33	142	99	0	0	0	0	0	...	201	190	54	43	255	255
78	255	255	211	173	199	163	0	...	0	0	91	80	255	255
97	170	151	0	0	0	0	0	...	33	30	173	162	255	255
66	255	232	156	93	124	63	0	...	0	0	20	13	255	255
129	255	255	106	109	57	74	0	...	46	46	157	150	255	255
37	212	176	8	3	0	0	0	...	0	0	136	120	255	255
76	255	244	162	131	136	113	0	...	200	191	0	0	255	255
47	175	139	3	0	0	0	0	...	0	0	126	114	255	255
124	174	146	38	44	24	37	0	...	80	76	193	183	255	255
76	255	194	64	29	8	0	0	...	0	0	37	30	255	255
134	255	255	190	167	150	137	0	...	82	75	103	94	255	255
75	224	180	0	0	0	0	0	...	60	52	83	72	255	255
73	254	195	148	107	151	108	0	...	66	54	82	69	255	255
99	119	104	0	0	0	0	0	...	12	11	186	175	255	255
81	255	255	216	154	209	145	0	...	68	56	41	29	255	255
103	250	212	0	6	0	0	0	...	24	19	143	131	255	255

Рисунок 2.4 - Формат зображення перед передачею алгоритму навчання

І вже в такому вигляді вони подаються на вхід алгоритму машинного навчання для тренування, або уже навченій моделі для класифікації.

3 Вибір алгоритму машинного навчання

Для вибору алгоритму навчання, який би підходив для даної роботи, було проведено тренування різних моделей та їх тестування за допомогою набору з 6250 фотографій із зображеннями різних видів котів та собак (рисунок 3.1).

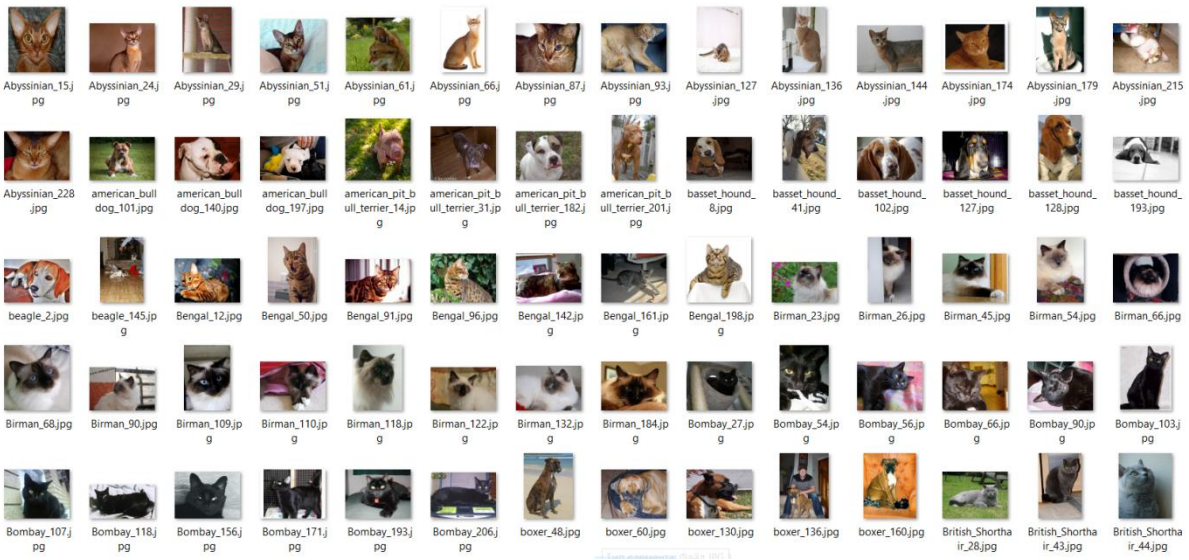

Рисунок 3.1 - Приклад тестових зображень

Далі розглянемо матриці невідповідностей для протестованих моделей та порахуємо їхню точність. Цей спосіб репрезентації точності моделі є досить хорошим, оскільки він дозволяє побачити не лише кількість правильних чи неправильних передбачить, але й зрозуміти характер помилок тим, що показує істинно позитивний (ліва верхня клітинка матриці), істинно негативний(права нижня), хибно позитивний(права верхня) та хибно негативний(ліва нижня) результати. Таким чином, отримуємо на головній діагоналі кількість правильних передбачень та кількість неправильних на побічній.

3.1 Метод опорних векторів(Support Vector Machines)

Першим я досліджував метод опорних векторів, оскільки він є досить універсальним і широко застосовується у задачах класифікації[15].

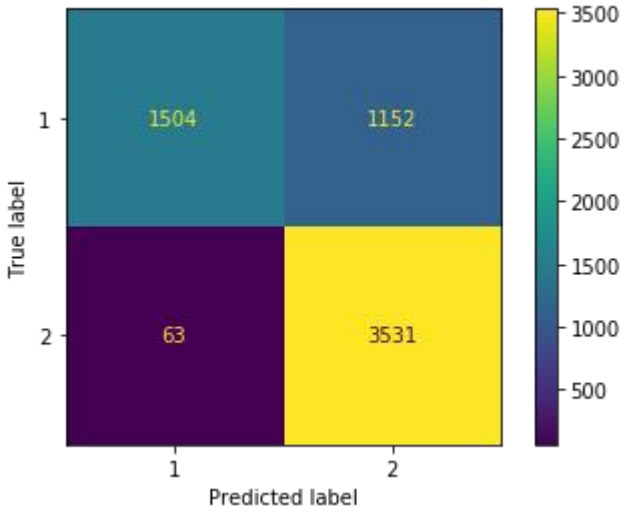

Рисунок 3.1.1 - Матриця невідповідності для методу опорних векторів

Можна побачити, що 5035 зображень було передбачено правильно, що означає, що дана модель дає точність у 80.6%.

3.2 Випадковий ліс (Random fores)

Після методу опорних векторів я досліджував роботу випадкового лісу. Цей спосіб машинного навчання також досить добре підходить для задач класифікації[16].

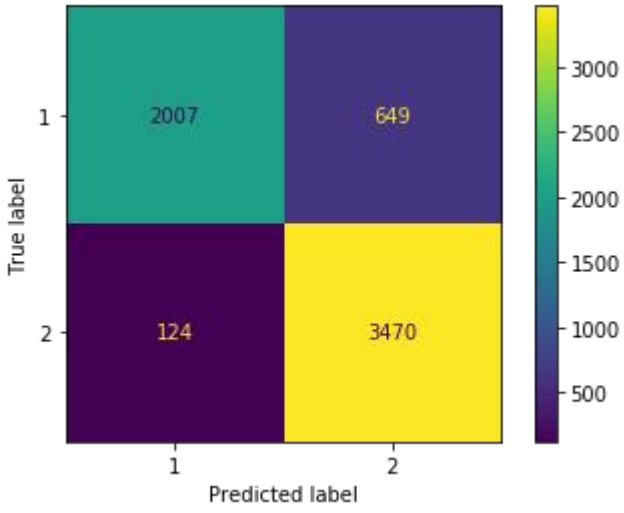

Рисунок 3.2.1 - Матриця невідповідності для випадкового лісу

При використанні даного методу точність моделі зросла на 7% і становить 87.6%. При цьому можна побачити, що кількість неправильно передбачених котів

зменшилась майже вдвічі у порівнянні із попередньою моделлю(права верхня клітинка матриці).

3.3 Перцептрон

Далі я розглядав ще один метод машинного навчання, який підходить для задач класифікації[17]. Мною були протестовані різні конфігурації перцептронів з одним прихованим шаром нейронів. Однією з найкращих була конфігурація, у якій прихований шар мав 256 нейронів, тобто чверть від кількості вхідних сигналів.

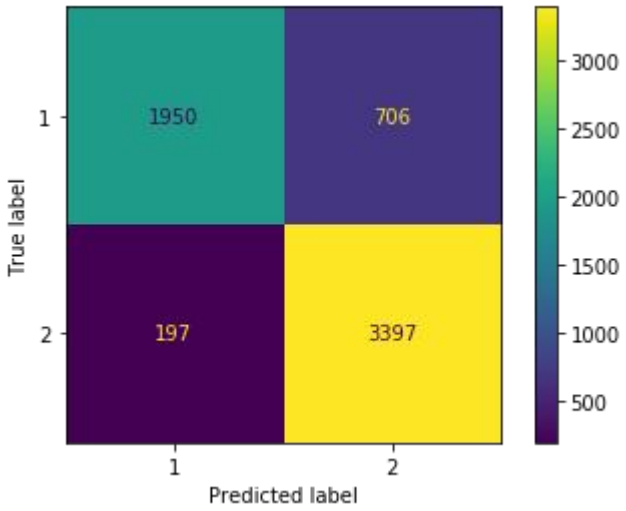

Рисунок 3.3.1 - Матриця невідповідності для тришарового перцептрона

Бачимо, що дана модель правильно передбачає 5347 зображень і відповідно має точність 86%.

3.4 Баєсівські класифікатори

На останок, мною були розглянуті різні баєсівські класифікатори. Вони також досить добре підходять для задач класифікації[18], крім того, методи цього класу відзначаються високою швидкістю навчання.

Почнемо з гаусівського наївного баєсу.

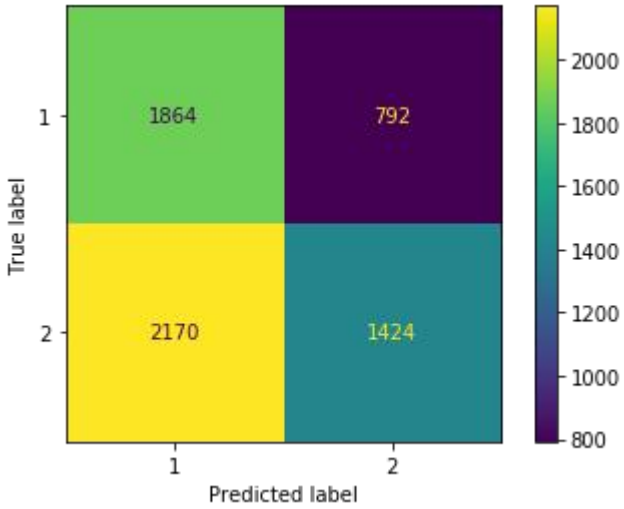

Рисунок 3.4.1 - Матриця невідповідності гаусівського наївного баєсу

Точність даної моделі склала 52%, що практично дорівнює вибору навмання. Далі розглянемо поліноміальний наївний баєс.

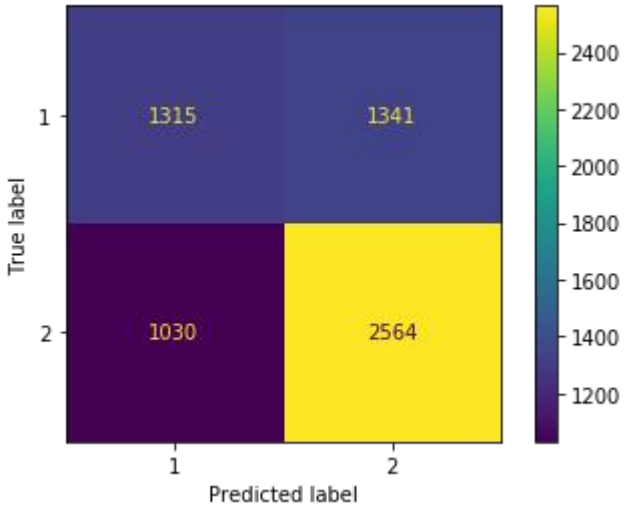

Рисунок 3.4.2 - Матриця невідповідності поліноміального наївного баєсу

Хоча точність цієї моделі є більшою ніж у попередньої й складає 62%, вона все одно є недостатньо хорошою, щоб її використовувати при вирішені поставленої задачі.

І на сам кінець, розглянемо наївний баєс Бернуллі.

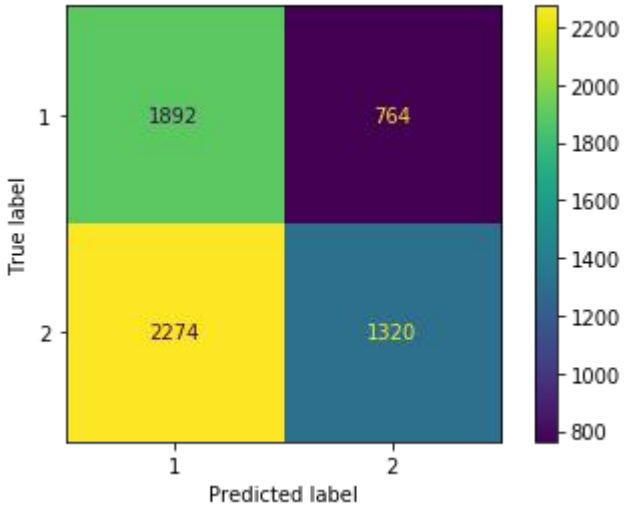

Рисунок 3.4.3 - Матриця невідповідності наївного баєсу Бернуллі

Дана модель показала найменшу точність з усіх розглянути - 51%, що як і у випадку з гаусівським наївним баєсом фактично є вибором навмання.

3.5 Висновки з дослідів

Як можна побачити, що найвищу точність має модель отримана за допомогою методу випадкового лісу, однак її навчання забирає значно більше часу у порівнянні з тришаровим перцептроном. Крім того, випадковий ліс не є гнучким, тобто при збільшенні кількості навчальних даних таку модель не можливо доповнити, в той самий час перцептрон можна легко дотренувати новими даними. Отже, попри те, що точність випадкового лісу на 1.6% більша, у даній роботі я вирішив використовувати тришаровий перцептрон.

4 Оптимізація навчання

Оскільки для тренування моделі та подальшої класифікації зображень використовується досить велика кількість фільтрів Ґабора (у даній роботі використовувалось 72 фільтри), то при видобуванні ознак кількість даних зростає пропорційно. При цьому часто стається так, що деякі з видобутих ознак майже не впливають на роботу моделі або, у гіршому випадку, зменшують її точність.

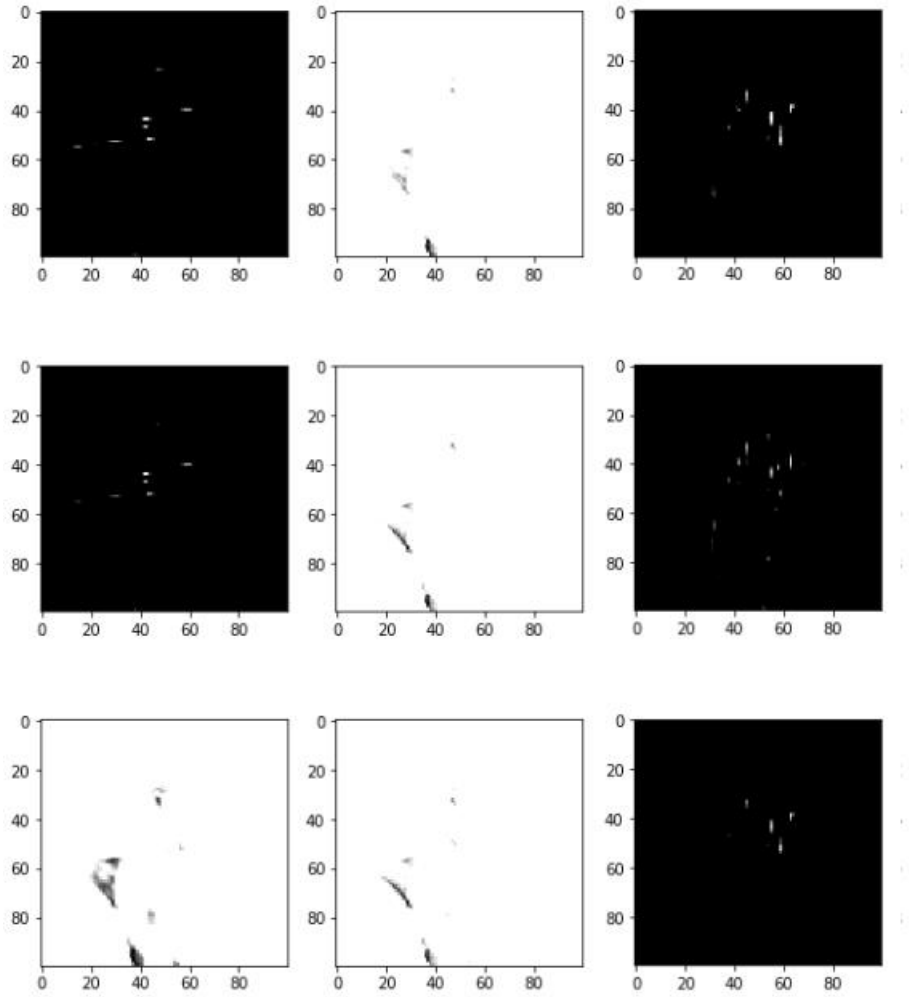

Рисунок 4.1 - Приклад видобутих ознак, які мають негативний вплив на навчання

Крім того, у будь-якому випадку, ці зайві ознаки збільшують час затрачений на навчання та обробку зображень. Саме тому виникає необхідність у певній оцінці фільтрів та відкиданні деяких з них ще до початку навчання.

4.1 Співвідношення Фішера

Необхідну оцінку, для вибору оптимального набору фільтрів можна отримати з відношення середньої різниці між класами до дисперсії за формулою:

$$J_k = \frac{\left\| m_{i,k} - m_{j,k} \right\|^2}{\left\| \sigma_{i,k} \right\|^2 + \left\| \sigma_{j,k} \right\|^2} \tag{4}$$

де $m_{i,k}$ та $m_{j,k}$ і $\sigma_{i,k}$ та $\sigma_{j,k}$- вектори середніх значень та стандартних відхилень для зображень, які належать *i*-тому і *j*-тому класам та оброблені *к*-тим фільтром. Дана формула називається співвідношенням Фішера[15].

Оцінка отримана з цієї формули показує на скільки добре ознаки, видобуті за допомогою певного фільтра Ґабора, дозволяють визначити до якого з класів належить задане зображення.

4.2 Використання співвідношення Фішера у навчанні

Для оптимізації навчання за допомогою формули (4) використаємо наступний алгоритм:

I. За допомогою процедури описаної у розділі 2 цієї роботи опрацюємо набір вхідних зображень та збережемо видобуті ознаки.
II. Обчислимо середнє значення та стандартне відхилення в кожному з класів, для кожного фільтра окремо. Далі обчислені значення підставимо у формулу (4) для отримання оцінок кожного з фільтрів. Результат цього кроку можна побачити на стовпчастій діаграмі на рис 4.2.1

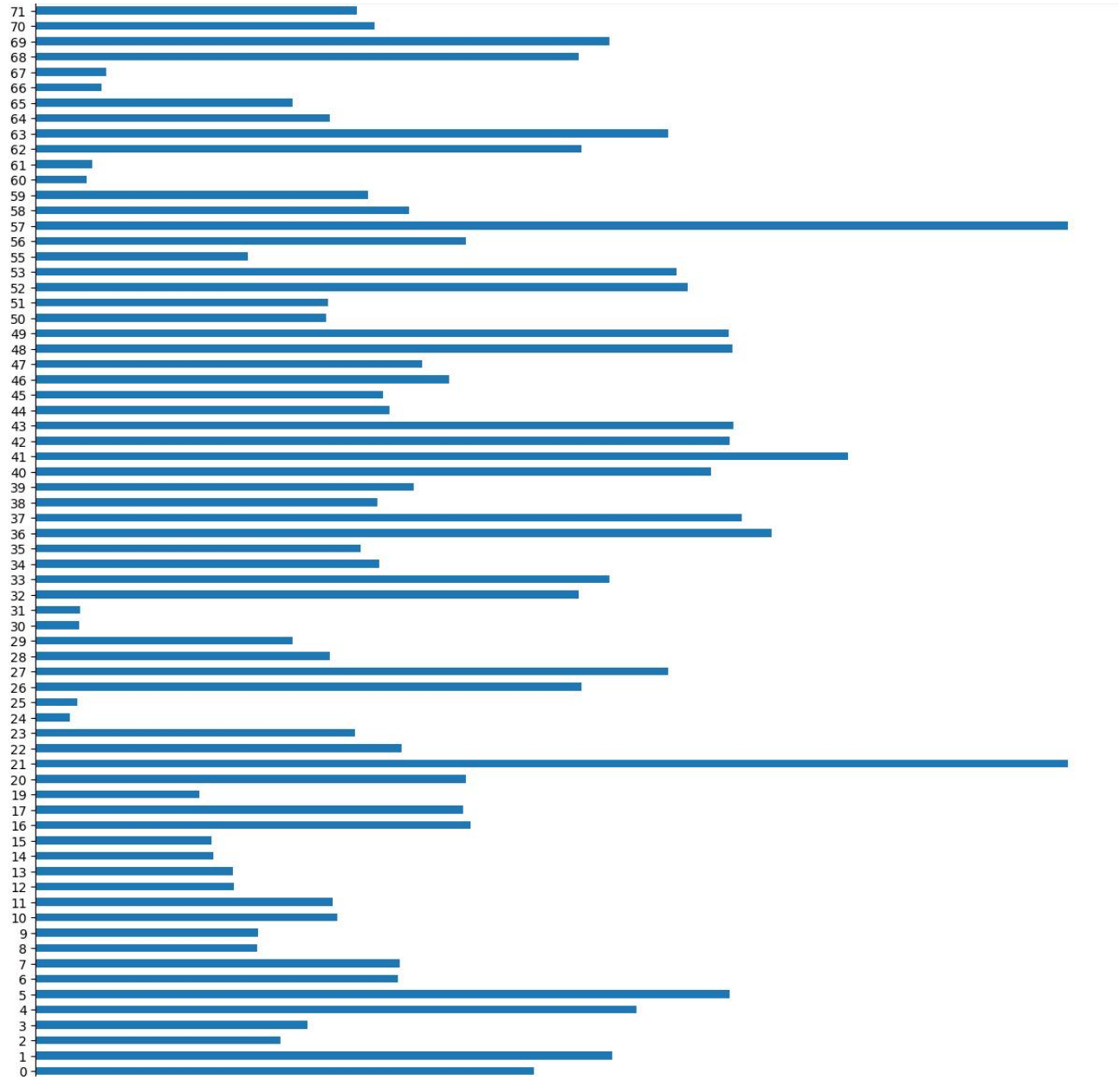

Рисунок 4.2.1 - Горизонтальна стовпчаста діаграма оцінок фільтрів

III. Наступним кроком потрібно відсортувати отримані результати у порядку спадання(рисунок 4.2.2). Після цього ми отримуємо порядок фільтрів від тих, які найкраще розділяють два класи, до тих, які роблять це дуже погано, або взагалі не розділяють.

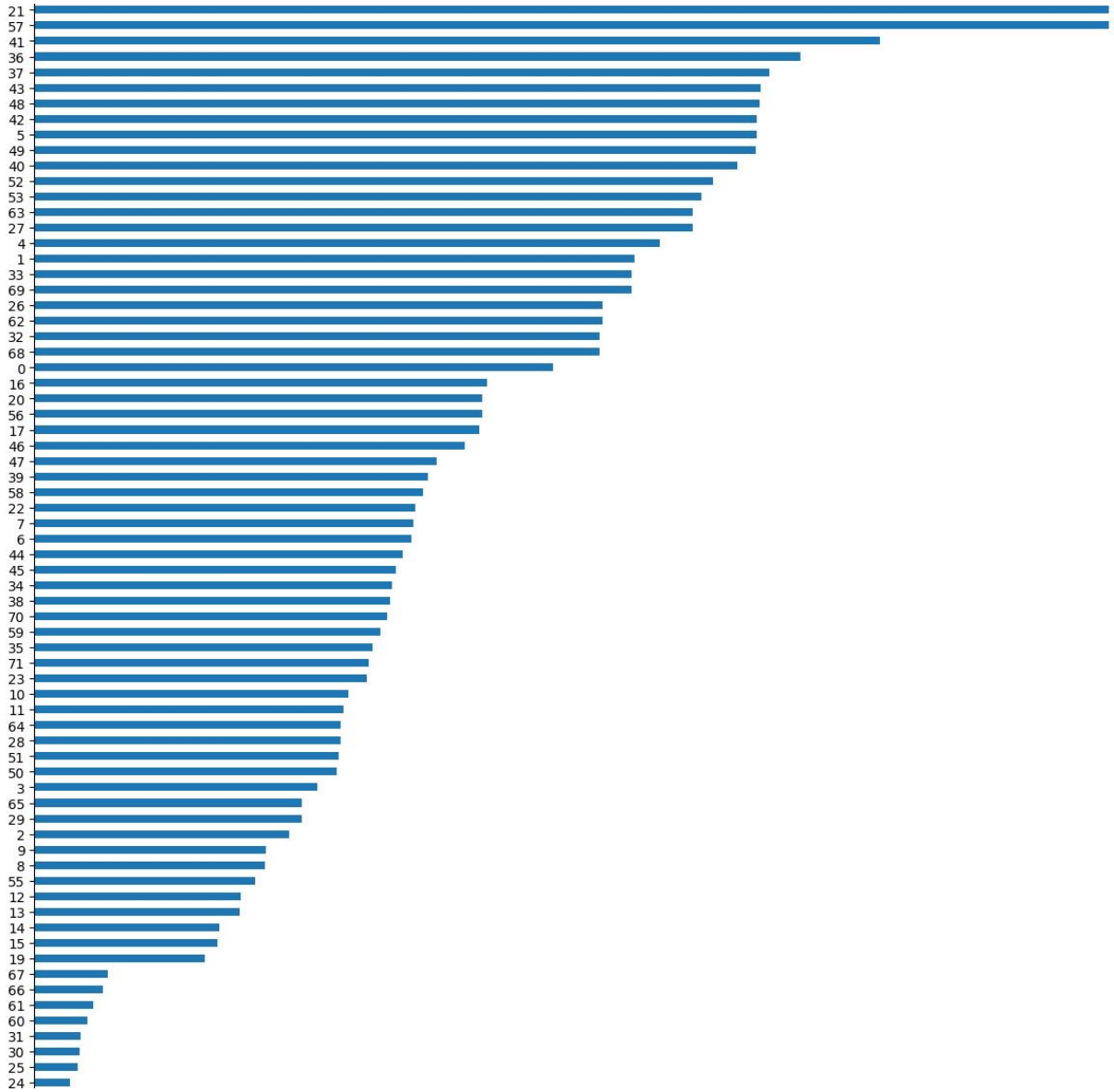

Рисунок 4.2.2 - Відсортована горизонтальна стовпчаста діаграма оцінок фільтрів

IV. Далі з набору усіх фільтрів обираємо найкращі, після цього застосовуємо їх до зображень та подаємо на вхід алгоритму машинного навчання, як це було описано раніше.

Далі було досліджено продуктивність навчання та точність моделі, при застосуванні вхідних даних, оброблених з використанням різної кількості фільтрів, обраних за допомогою вищенаведеного алгоритму. Отримані після експериментів метрики для різної кількості фільтрів подано в таблиці 1.

Таблиця 1 - Результати навчання з різною кількістю фільтрів

Кількість фільтрів	Час застосування фільтрів (с)	Час навчання моделі (хв)	Точність моделі
72	237.0524764060974	486.1	86%
60	193.73135495185852	356.3	86.5%
50	172.29346299171448	231.2	87.3%
30	121.63584637641907	86.9	88.9%
20	98.50894808769226	33.44	87.4%
10	86.82412314414978	17.2	83.8%

З цієї таблиці можна побачити, що з відкиданням фільтрів з низькою оцінкою починає збільшуватись точність моделі. Однак, можна помітити, що при використанні 20 фільтрів або менше точність починає знижуватись та падає до 83.8% (якщо використовувати 10 фільтрів). Такий результат пов'язанийз тим, що зі зменшенням кількості фільтрів, кількість видобутих ознак стає недостатньою для точної класифікації зображень.

Також з таблиці 1 можна зробити висновок, що в плані точності моделі, оптимальна кількість фільтрів для обраного класу зображень становить 30. При добуванні ознак тридцятьма фільтрами точність моделі зростає на 3%, крім цього, час затрачений на застосування цих фільтрів до зображень та їх підготовку до навчання зменшується майже вдвічі.

Однак, чи не найважливішим є час затрачений на навчання моделі. Можна побачити, що у той час, як для навчання моделі з використанням усіх 72-х фільтрів знадобилось більше ніж вісім годин, час затрачений на навчання моделі при використанні 30 фільтрів становить приблизно півтори години. Таким чином відбір фільтрів за допомогою формули (4) не лише допоміг покращити точність моделі, а й призвів до разючого (більш ніж у 5 разів) прискорення навчання.

Висновки

Для досягнення поставленої мети у цій роботі було проаналізовано фільтри Ґабора, досліджено вплив їх параметрів на самі фільтрі та на зображення і створено банк фільтрів для видобування ознак із картинок.

Також було розглянуто різні алгоритми машинного навчання, такі як метод опорних векторів, випадковий ліс, багатошаровий перцептрон і наївні баєсівські класифікатори та протестовано їхню ефективність, при тренуванні за допомогою набору ознак отриманих з використанням набору різних фільтрів Ґабора та було обрано модель машинного навчання, яка найбільше підходить для даної проблеми.

Крім того, було оптимізовано навчання обраної моделі, шляхом зменшення кількості фільтрів, які застосовуються. Це зменшення відбулося на основі оцінки отриманої зі співвідношення Фішера. В результаті вдалося досягти значного прискорення навчання та покращення точності обраної моделі.

Як результат виконання даної роботи, було створено систему, яка на основі набору фільтрів Ґабора та за допомогою тришарового перцептрона, дозволяє вибрати фото, подібні до наперед заданих, із великого набору зображень. А також було оптимізовано процеси навчання та класифікації для цієї моделі машинного навчання.

Підсумовуючи, можна зробити висновок, що ознаки видобуті із зображення за допомогою фільтрів Ґабора справді досить таки добре підходять для навчання класифікаторів, хоча і з’являється певна надлишковість у вигляді нерелевантних ознак, а також зростає час затрачений на здійснення операцій над зображеннями та час затрачений на навчання. На додачу, у результаті досліджень виявилось, що тришаровий перцептрон дуже добре підійшов для розв’язання поставленої задачі, попри те, що цей метод машинного навчання є одним із найпримітивніших. Крім того, вдалось з'ясувати, що за допомогою оптимального вибору набору фільтрів, можна майже повністю нівелювати негативний ефект від надлишковості

згенерованих фільтрів Ґабора, оскільки вона фактично прибирається. І як з'ясувалось оцінка отримана зі співвідношення Фішера чудово для цього підходить.

Список використаних джерел

1. Brahmbhatt S. Introduction to computer vision and opencv [Electronic resource] / Samarth Brahmbhatt // Practical OpenCV. – Berkeley, CA, 2013. – P. 3–5. – Mode of access: https://doi.org/10.1007/978-1-4302-6080-6_1
2. Computer vision / Linda G. Shapiro [et al.]. – [S. l.] : Prentice Hall, 2001. – 580 p.
3. Olshausen B. A. Emergence of simple-cell receptive field properties by learning a sparse code for natural images [Electronic resource] / Bruno A. Olshausen, David J. Field // Nature. – 1996. – Vol. 381, no. 6583. – P. 607–609. – Mode of access: https://doi.org/10.1038/381607a0
4. Contributors to Wikimedia projects. Gabor filter - Wikipedia [Electronic resource] / Contributors to Wikimedia projects // Wikipedia, the free encyclopedia. – [S. l.], 2005. – Mode of access: https://en.wikipedia.org/wiki/Gabor_filter.
5. Go H.-J. Iris recognition using the 2-D gabor filter [Electronic resource] / Hyoun-Joo Go, Dae-Jong Lee, Myung-Geun Chun // Journal of korean institute of intelligent systems. – 2003. – Vol. 13, no. 6. – P. 716–721. – Mode of access: https://doi.org/10.5391/jkiis.2003.13.6.716.
6. Jain A. K. Unsupervised texture segmentation using Gabor filters [Electronic resource] / Anil K. Jain, Farshid Farrokhnia // Pattern recognition. – 1991. – Vol. 24, no. 12. – P. 1167–1186. – Mode of access: https://doi.org/10.1016/0031-3203(91)90143-s.
7. Haghighat M. Identification using encrypted biometrics [Electronic resource] / Mohammad Haghighat, Saman Zonouz, Mohamed Abdel-Mottaleb // Computer analysis of images and patterns. – Berlin, Heidelberg, 2013. – P. 440–448. – Mode of access: https://doi.org/10.1007/978-3-642-40246-3_55.
8. Sanger T. D. Stereo disparity computation using Gabor filters [Electronic resource] / T. D. Sanger // Biological cybernetics. – 1988. – Vol. 59, no. 6. – P. 405–418. – Mode of access: https://doi.org/10.1007/bf00336114.

9. Nonlinear model of neural responses in cat visual cortex [Electronic resource] // Computational models of visual processing. – [S. l.], 1991. – Mode of access: https://doi.org/10.7551/mitpress/2002.003.0014.
10. K. Ali H. Image subset selection using gabor filters and neural networks [Electronic resource] / Heider K. Ali, Anthony Whitehead // The international journal of multimedia & its applications. – 2015. – Vol. 7, no. 2. – P. 43–55. – Mode of access: https://doi.org/10.5121/ijma.2015.7204.
11. Kapur S. Computer Vision with Python 3: Use the power of Python for real-time image processing and analysis / Saurabh Kapur. – [S. l.] : Packt Publishing, 2017. – 206 p.
12. Multi-categorical deep learning neural network to classify retinal images: a pilot study employing small database [Electronic resource] / Joon Yul Choi [et al.] // Plos one. – 2017. – Vol. 12, no. 11. – P. e0187336. – Mode of access: https://doi.org/10.1371/journal.pone.0187336.
13. OpenCV: OpenCV modules [Electronic resource] // OpenCV documentation index. – Mode of access: https://docs.opencv.org/4.x/.
14. Sarle W. S. Part 1 of 7: introduction. / Warren S. Sarle // Neural network FAQ. – Cary, NC, USA, 2002
15. Cortes C. Support-vector networks [Electronic resource] / Corinna Cortes, Vladimir Vapnik // Machine learning. – 1995. – Vol. 20, no. 3. – P. 273–297. – Mode of access: https://doi.org/10.1007/bf00994018.
16. Breiman L. / Leo Breiman // Machine learning. – 2001. – Vol. 45, no. 1. – P. 5–32. – Mode of access: https://doi.org/10.1023/a:1010933404324.
17. Bello M. G. Enhanced training algorithms, and integrated training/architecture selection for multilayer perceptron networks [Electronic resource] / M. G. Bello // IEEE transactions on neural networks. – 1992. – Vol. 3, no. 6. – P. 864–875. – Mode of access: https://doi.org/10.1109/72.165589.
18. Ashfaq T. Classification of hand gestures using gabor filter with bayesian and naïve bayes classifier [Electronic resource] / Tahira Ashfaq, Khurram Khurshid //

International journal of advanced computer science and applications. – 2016. – Vol. 7, no. 3. – Mode of access: https://doi.org/10.14569/ijacsa.2016.070340.

19. Sung J. A Bayesian network classifier and hierarchical Gabor features for handwritten numeral recognition [Electronic resource] / Jaemo Sung, Sung-Yang Bang, Seungjin Choi // Pattern recognition letters. – 2006. – Vol. 27, no. 1. – P. 66–75. – Mode of access: https://doi.org/10.1016/j.patrec.2005.07.003.
20. Lohninger H. Fundamentals of statistics. / H. Lohninger. – [S. l.] : Epina Bookshelf.
21. Trappenberg T. P. Machine learning with sklearn [Electronic resource] / Thomas P. Trappenberg // Fundamentals of machine learning. – [S. l.], 2019. – P. 38–65. – Mode of access: https://doi.org/10.1093/oso/9780198828044.003.0003.
22. VanderPlas J. Python data science handbook: essential tools for working with data / Jake VanderPlas ; ed. by D. Schanafelt. – [S. l.] : O'Reilly Media, 2016. – 548 p.
23. Selection of Gabor filters for improved texture feature extraction [Electronic resource] / Weitao Li [et al.] // 2010 17th IEEE international conference on image processing (ICIP 2010), Hong Kong, Hong Kong, 26–29 September 2010. – [S. l.], 2010. – Mode of access: https://doi.org/10.1109/icip.2010.5653278.
24. Dat T. H. Feature selection based on fisher ratio and mutual information analyses for robust brain computer interface [Electronic resource] / Tran Huy Dat, Cuntai Guan // 2007 IEEE international conference on acoustics, speech and signal processing - ICASSP '07, Honolulu, HI, USA, 15–20 April 2007. – [S. l.], 2007. – Mode of access: https://doi.org/10.1109/icassp.2007.366685.
25. Duda R. O. Pattern classification and scene analysis / Richard O. Duda. – New York : Wiley, 1973. – 482 p.

Додаток А. Код для попередньої обробки зображень та їх форматування для навчання

```
def preprocess(img,size=(32,32)):
    gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
    resized=cv.resize(gray,size)
    clahe = cv.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
    normalized = clahe.apply(resized)
    return normalized

def appply_filter(img,kernels,lables):
    df=pd.DataFrame()
    df['Original Image']=img.reshape(-1)
    raw_features=[]
    for kernel,label in zip(kernels,lables):
        fimg = cv.filter2D(img,cv.CV_8UC3,kernel)
        raw_features.append(fimg)
        df[label]=fimg.reshape(-1)
    return df,raw_features

def form_data(frame,kernels=None,lables=None,size=(32,32)):
    if kernels==None or lables==None:
        kernels,lables=generatte_filters()
    X=[]
    Y=[]
    for i in range(frame.shape[0]):
        path=os.path.relpath(frame.iloc[i]['file'])
        im=cv.imread(path)
        try:
            len(im)
        except:
            print(path)
            continue
        cls=[int(frame.iloc[i]['class'])]
        proc_im=preprocess(im,size)
        data_f,raw_features=appply_filter(proc_im,kernels,lables)
        feat_array=np.array(data_f.to_numpy().T)
        X.append(feat_array)
        Y.append(cls*data_f.shape[1])
    X=np.array(X)
    Y=np.array(Y)
    X=X.reshape(-1,X.shape[-1])
    Y=Y.reshape(-1)
    return X,Y
```

Додаток Б. Код для оцінки та вибору фільтрів

```
def calculate_score(X,Y,norm_type=np.inf):
    X=np.array(X)
    Y=np.array(Y)
    i=0
    scores=[]
    for k in range(0,X.shape[1]):
        i=i+1
        m1k=X[:,k][Y==0].mean(0)
        m2k=X[:,k][Y==1].mean(0)
        sig1k=X[:,k][Y==0].std(0)
        sig2k=X[:,k][Y==1].std(0)
        j=(norm(m1k-m2k,norm_type)**2)/(norm(sig1k**2,norm_type)+norm(sig2k**2,norm_type))
        scores.append(j)
    return scores

def select_filters(X,Y,all_kernels,all_lables,number_of_filters =30):
    scores=calculate_score(X,Y);
    ScoreBoard=pd.DataFrame(scores,[i for i in range(0,len(all_kernels))])
    ScoreBoard=ScoreBoard.dropna()
    ScoreBoard=ScoreBoard.sort_values(by=0)
    selected_kernels=[all_kernels[i] for i in ScoreBoard.iloc[-number_of_filters:].index]
    selsected_lables=[all_lables[i] for i in ScoreBoard.iloc[-number_of_filters:].index]
    return selected_kernels,selsected_lables
```