UDC 004.4: 004.4 DOi 10.30837/ bi.2024.2(101).12

V. I. Dubrovin[1], O. V. Petunin[2]

[1]Candidate of Technical Sciences, Professor, Department of Software,
National University "Zaporizhzhia Polytechnic", Zaporizhzhia, Ukraine,
e-mail: vdubrovin@gmail.com, ORCID ID: 0000-0002-0848-8202

[2]master's degree student of the department of software tools National University "Zaporizhzhia Polytechnic",
Zaporizhzhia, Ukraine, e-mail: zephyr.42.foton@gmail.com, ORCID ID: 0009-0008-6516-7561

GLAUCOMA DIAGNOSTICS USING MACHINE LEARNING METHODS

The research focuses on eusage of machinel earning algorithms in glaucoma diagnostics. The objective is to analyze and compare various machine learning algorithms by constructing classification systems that verify glaucoma in ICT photos. The study involves extracting feature from photos, classifiying them using this features and evaluating the effectivenes of methods. This approach provides insights into creation of automated glaucoma diagnostic system, contributing to faster and safer medical process.

GLAUCOMA, DIAGNOSIS, NEAREST NEIGHBORS METHOD, NAIVE BAYES, FEATURES, MACHINE LEARNING, EVALUATION METRICS, CLASSIFICATION, SAMPLING

В.І. Дубровін, О.В. Петунін. Діагностика глаукоми методами машинного навчання. Дослідження зосереджено на використанні алгоритмів машинного заробітку в діагностиці глаукоми. Мета полягає в тому, щоб проаналізувати та порівняти різні алгоритми машинного навчання шляхом побудови систем класифікації, які перевіряють глаукому на фотографіях ІКТ. Дослідження передбачає виділення ознак із фотографій, їх класифікацію за цими ознаками та оцінку ефективності методів. Такий підхід дає можливість зрозуміти створення автоматизованої системи діагностики глаукоми, що сприяє більш швидкому та безпечному процесу лікування.

ГЛАУКОМА, ДІАГНОЗ, МЕТОД НАЙБЛИЖЧИХ СУСІДІВ, НАЇВНИЙ БАЙЄС, ОСОБЛИВОСТІ, МАШИННЕ НАВЧАННЯ, МЕТРИКИ ОЦІНЮВАННЯ, КЛАСИФІКАЦІЯ, ВИБІРКА

1. Introduction

In today's era of rapid advancements in medicine and increasing demand for precise diagnostics, automation has become a fundamental aspect of healthcare innovation. The ability to automate diagnostic systems not only improves accuracy and efficiency but also gives a competitive advantage in the healthcare market. As clinical needs increase, achieving reliable and swift diagnostic performance is non-negotiable.

Glaucoma is an incurable disease that causes vision loss and is the second leading cause of blindness in the world [1]. To detect glaucoma, experts use several imaging techniques, including confocal scanning laser ophthalmoscopy (CSLO), Heidelberg retinal tomography (HRT), optical coherence tomography (OCT), and fundus imaging [3]. Based on the imaging technique, several features of the retinal structure, such as the optic nerve head (ONH), the cup, peripapillary atrophy, and the retinal nerve fiber layer, need to be observed to detect glaucoma. In the fundus image, the ONH is a bright and round area, and inside the ONH is a smaller round area called the cup. Peripapillary atrophy appears as a crescent that overlaps with the area outside the ONH. The retinal nerve fiber layer is also located outside the ONH, which has a white striped structure [3; 4].

Machine learning, a transformative intelligence technology, has reshaped the world of diagnostics. Its capacity to process and analyze vast amounts of medical data, including images and patient records, is making it an important tool in addressing complex health problems. Glaucoma, a leading cause of irreversible blindness worldwide, is one such condition where early and accurate diagnosis can significantly reduce the risk of vision loss. Building and optimizing a functional ML model for glaucoma detection and ensuring they deliver consistent results across diverse clinical scenarios is crucial for succes in a world of modern medicine.

This article explores the role of machine learning in glaucoma diagnosis, with a particular focus on difference in efficiency between various methods of classsification. Among the myriad factors influencing model efficacy, data processing and feature extraction stand out as pivotal. These steps are crucial to ensuring the model accurately identifies early signs of glaucoma, such as changes in the optic nerve head and retinal nerve fiber layer, from diagnostic imaging modalities of optical coherence tomography (OCT) [5;6].

The findings presented in this article are important for technologists aiming to create their diagnostic tools. They underscore the importance of balancing technical performance with clinical applicability, setting a new benchmark for excellence in medical innovation. Finally, this research contributes to the broader field of healthcare technology, opening new pathways for improving patient outcomes and advancing global health standards

2. Why glaucoma diagnostic is important

Diagnosing glaucoma accurately and early is critical for several reasons, as it directly impacts patient's outcome.Timely detection is the requierment of effective glaucoma management [7]. Early diagnosis enables proactive intervention, which can significantly slow the progression of the disease and preserve vision:

– Preventing irreversible damage: Glaucoma in early stages progresses without noticeable symptoms until significant vision loss occurs. Early detection is the only way to intervene before permanent damage sets in.

– Avoiding advanced stages: Advanced glaucoma often requires more invasive and costly treatments, such as surgery. Early intervention reduces the likelihood of reaching this.

Effective glaucoma diagnostics also play a key role in enhancing healthcare efficiency:

– Reducing the burden on specialists: Automated or semi-automated diagnostic tools powered by machine learning can assist ophthalmologists, allowing them to focus on complex cases while routine screenings are handled faster.

– Improving accessibility: Many regions lack access to specialists or special equipment. Automated diagnostic methods can help bridge this gap, especially in impoverished areas.

– Cost savings: The automation helps save on salaries for specialists, while early diagnosis reduces the long-term healthcare costs required for managing advanced glaucoma and its complications.

From a societal perspective, robust diagnostic systems have far-reaching benefits:

– Reducing the global burden of blindness: Glaucoma is a leading cause of preventable blindness worldwide. Early and accurate diagnosis can significantly lower its prevalence.

– Promoting health equity: Accessible and efficient diagnostic tools ensure that even individuals in resource-limited settings can benefit from early detection and treatment.

– Enhancing public health outcomes: Early interventions reduce disability rates and associated social and economic costs.

Lastly, advancements in glaucoma diagnostics contribute to broader innovations in ophthalmology and medical technology:

– Data-driven insights: Machine learning-powered diagnostic systems generate valuable data that can inform better treatment protocols and personalized care.

– Encouraging innovation: Research and development in glaucoma diagnostics often pave the way for improved diagnostic methods in other medical fields.

Automated glaucoma diagnostics is not just about detecting a disease; it's about strengthening healthcare systems, and contributing to the global fight against blindness. By prioritizing medical innovations, the scientific community can ensure that people receive the care they need when it matters most.

3. Glaucoma diagnostics with machine learning

In the context of glaucoma diagnosis, precision and reliability are usually achieved with two methods:

– Diagnostic Accuracy –precise identification of glaucoma signs ensures correct differentiation between healthy and ill eyes. Accurate diagnostics reduce false positives and negatives, allowing for sureness in treating patients.

– Time of Diagnosis – the speed at which mages are analyzed is crucial for treatment effectiveness. Rapid diagnostic methods provide a critical window for intervention to preserve vision and prevent progression.

Ensuring accurate analysis and feature extraction is essential for improving glaucoma diagnostic outcomes and advancing eye care [12].

Several factors influence the effectiveness of glaucoma diagnostics. Here are the primary considerations:

– Localization of regions of interest (ROI): The optic nerve head must be accurately separated from the images to ensure correct analysis. Improper localization can devaluate the diagnosis accuracy [13; 14].

– Image quality and resolution: High-quality fundus images and OCT scans ease detecting subtle structural changes indicative of glaucoma. Poor image resolution can obscure key details and lead to diagnostic errors.

– Algorithm robustness: Machine learning algorithms must be robust enough to handle variations in imaging data, such as differences in illumination, contrast, or anatomical variability.

– Integration of multiple imaging modalities: Combining data from various techniques, like OCT, fundus photography, and visual field tests, enhances accuracy by providing an informative view of retinal health [9].

Considering these factors, it is crucial to apply effective techniques and tools to improve the diagnostic process. Some key approaches include:

– Convolutional Neural Networks (CNNs);
– Support Vector Machines (SVMs):
– Unsupervised learning techniques;
– Classification algorithms;
– Ensemble learning models;.
– Deep learning models for segmentation;
– Reinforcement learning;
– Transfer learning.

By focusing on these approaches, researchers and clinicians can develop and deploy computer-aided diagnostic (CAD) tools. These tools have the potential to change diagnosis of glaucoma, making it accessible to populations of poor countries and giving a chance for early intervention. By prioritizing such research scientists can solve one of the world's leading causes of blindness and improve patient outcomes.

4. Selection of classification methods

To conduct a study of glaucoma diagnostic methods, a decision was made to utilize the k-Nearest Neighbors (k-NN) and Naive Bayes classifications algorithms for several reasons. These choices are explained by the specifics of building a reliable diagnostic system:

– Simplification of Model Complexity:

Both k-NN and Naive Bayes are relatively simple yet effective machine learning algorithms. k-NN avoids assumptions about data distribution, and Naive Bayes uses straightforward probability calculations, making them ideal for initial evaluations. This simplicity ensures transparency in their operation, enabling clear interpretation of results without introducing unnecessary complexity [10].

– Measurement and Comparison Capability:

For glaucoma diagnosis, each algorithm's performance is measured using standard metrics such as sensitivity,specifity and accuracy. The use of k-NN and Naive Bayes facilitates direct comparison of their ability to classify eye images correctly. Each classifier serves as a separate diagnostic approach, enabling the analysis of their strengths and weaknesses separately from one another [11].

– Convenience for Results Analysis:

Both classifiers provide clear outputs that allow for straightforward assessment of diagnostic performance. k-NN assigns class labels based on proximity, and Naive Bayes calculates posterior probabilities for each class. This enables the identification of which features or parameters contribute most to diagnostic accuracy, aiding in the feature selection and preprocessing techniques.

Given these factors, the combination of k-NN and Naive Bayes is an optimal choice for glaucoma diagnostic research [15]. These methods provide complementary simplicity and performance, making them effective tools for assessing diagnostic system efficacy while maintaining experimental clarity and reliability.

5. Software development for conducting research

To conduct this research, a specialized program was developed for processing the image features by isolating distinct mathematical characteristics derived from pixel-level data. The program's workflow is organized into a series of structured steps that allow for the completion of image classification tasks. Below is an overview of the program's work process and its corresponding modules.

When the program is launched, users can access options to select a dataset and initiate analyzing tasks.

Image Selection and Preprocessing.

When the user selects a folder containing images, the program takes each file in the folder. The ImageCrop module then crops the area of interest within each image by converting the image to HSV format and isolating the the vakue layer. Every pixel that has value below 97% of the brightest pixel is removed, turning image into binary. The biggest is found among remaining pixels and it borders are overlaid with original image, turning it into zone of interest. The code (Fig. 1) for cropping the image and the result (Fig. 2). of it is shown below.

```
img = cv.imread(file, cv.IMREAD_COLOR)
if img is None:
raise ValueError("Image not found or unable to load.")
#show_image(img, "Original Image")
lab = cv.cvtColor(img, cv.COLOR_BGR2LAB)
l_channel, a, b = cv.split(lab)
#show_image(l_channel)
clahe = cv.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))
cl = clahe.apply(l_channel)
#show_image(cl)
limg = cv.merge((cl, a, b))
enhanced_img = cv.cvtColor(limg, cv.COLOR_LAB2BGR)
#show_image(enhanced_img, "Enhanced Image")
# Convert to HSV for contour detection
hsv = cv.cvtColor(img, cv.COLOR_BGR2HSV)
h, s, v = cv.split(hsv)
#show_image(v)
max_val = np.max(v)
per = 0.97
_, th = cv.threshold(v, max_val * per, 255, cv.THRESH_BINARY)
#show_image(th, "Thresholded Image")
contours, hierarchy = cv.findContours(th, cv.RETR_EXTERNAL, cv.CHAIN_APPROX_SIMPLE)
max_area = 0
largest_contour = None
for cnt in contours:
area = cv.contourArea(cnt)
if area > max_area:
max_area = area
largest_contour = cnt
contour_image = img.copy()
if largest_contour is not None:
cv.drawContours(contour_image, [largest_contour], -1, (0, 255, 0), 2) # Green contour
#show_image(contour_image, "Contours on Image")
x, y, w, h = cv.boundingRect(largest_contour)
roi = img[y:y+h, x:x+w]
#show_image(roi, "Region of Interest")
return roi
else:
print("No contours found.")
return None
```

Fig. 1. Cropping of area of interest

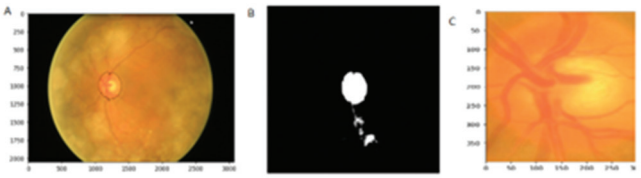

Fig. 2. a) Original image, b) Processed Thresholded Value channel, c) Cropped area of interest

The cropped area of the image is then passed to the FeatureExtraction module, where image features are extracted. In this research it includes the mean, standard deviation, symmetry, and skewness. These features are derived from pixel intensity values of images and collect data for classifications. Below is the code (Fig. 3) for the function that does it.

```
b,g,r=cv.split(img)

e=0
count=0

for pixel in g:
  for pixel2 in pixel:
    e+=pixel2
    count+=1
u=e/count
e=0
for pixel in g:
  for pixel2 in pixel:
    e+=(math.pow(pixel2-u,2))
sigma=e/count
r=1-(1/(1+sigma))

row=[]
for pixel in g:
  for pixel2 in pixel:
    row.append(pixel2)
sk=(skew(row, axis=0, bias=True))
ks=(kurtosis(row, axis=0, bias=True))
std=math.sqrt(sigma)
u=round(u,4)
std=round(std,4)
r=round(r,4)
sk=round(sk,4)
ks=round(ks,4)
print(u,std,r,sk,ks)
return u,std,r,sk,ks
```

Fig. 3. Feature extraction

Training the Nearest Neighbor System

Using the HandleKnn module, users can train a nearest neighbors' classification system. When training is initiated, the program extracts feature from the provided training images and stores them in Json format. Each feature set is associated with a specific class label for later reference during testing. The distance matrix is calculated using the Euclidean distance (d), which is found using formula (1) below.

$$d = \sqrt{\sum_{i=1}^{c}(x_i - y_i)^2}\ , \qquad (1)$$

where i is the recurrence index, which is the number of features. The feature matrix of the sample data is expressed

by x_i, while for the test data it is expressed by y_i. The code (Fig. 4). For it is shown below.

```
folder_path = filedialog.askdirectory()
neg = []
i = 0
for filename in os.scandir(folder_path):
    i += 1
    if filename.is_file():
        img = cropImg(filename.path)
        ex = extract(img)
        neg.append(ex)
    if i == 50:
        break

with open('Studies/knn.json') as f:
    data = json.load(f)

data[name] = neg
with open('Studies/knn.json', 'w') as file:
    json.dump(data, file, indent=4)
```

Fig. 4. KNN training

In the same way users can train a naïve bayes system. The naive bayes module calculates probabilities for each class based on extracted features. A Gaussian probability distribution is calculated for each feature, which is found using formula (2) below.

$$P(c|x) = \frac{P(x|c) * P(c)}{P(x)}, \qquad (2)$$

where, *P(c|x)* is the posterior probability of the class *(c, target)* with a given predictor *(x, attributes), P(c)* is the prior probability of the class, *P(x|c)* is the probability that is the probability of a given class of the predictor, *P(x)* is the prior probability of the predictor.

These probabilities are stored for use in the testing phase. The code (Fig. 5) for training it is shown below.

```
folder_path = filedialog.askdirectory()
if not folder_path:
    print("No folder selected.")
    return

neg = []
for i, filename in enumerate(os.scandir(folder_path)):
    if i >= 50:
        break
    if filename.is_file():
        img = cropImg(filename.path)
        ex = extract(img)
        neg.append(ex)

# Calculate mean and variance for each feature
stats = []
for i in range(len(neg[0])):
    col = [entry[i] for entry in neg]
    mean = sum(col) / len(col)
    variance = math.sqrt(sum([(x - mean) ** 2 for x in col]) / (len(col) - 1))
    stats.append([mean, variance, len(col)])

# Update JSON file
try:
    with open('Studies/nb.json', 'r') as f:
        data = json.load(f)
except (FileNotFoundError, json.JSONDecodeError):
    data = {}

data[name] = stats

with open('Studies/nb.json', 'w') as f:
    json.dump(data, f, indent=4)
```

Fig. 5. Naïve Bayes training

After training in either of tests, the program automatically saves the values of extracted features and Gaussian probabilities, rewriting existing records in its files.

Testing the K-Nearest Neighbor System.

Following this the program allows to test the trained models with a new set of images. In KNN The program accepts a folder of test images, extracts features, and compares them against the training data using the k-NN algorithm. The class of the nearest matches (default k=5) is assigned to the test images. The algorithm (Fig. 6). is shown in the image below.

```
folder_path = filedialog.askdirectory()
    f = open('Studies/knn.json')

    data = json.load(f)

    f.close()
    negc=0
    posc=0
    stats=[]
    objs={}
    results = {}
    for filename in os.scandir(folder_path):
        if filename.is_file():

            img=cropImg(filename.path)

            ex=extract(img)
            stats.append(ex)
            objs[filename]=""
            neg.pos=[].[]
            for point in data["neg"]:
                dist=0
                for i in range(5):
                        dist+=math.pow(ex[i]-
point[i],2)
                dist=round(math.sqrt(dist),4)
                neg.append(dist)
            for point in data["pos"]:
                dist=0
                for i in range(5):
                        dist+=math.pow(ex[i]-
point[i],2)
                dist=round(math.sqrt(dist),4)
                pos.append(dist)
            countn,countp=0,0
            pos.sort()
            neg.sort()

            for i in range(10):
                if(pos[0]>neg[0]):
                    countn+=1
                    neg.pop()
                else:
                    countp+=1
                    pos.pop()

            if(countn<countp):
                objs[filename]="negative"
                negc+=1
            else:
                objs[filename]="positive"
                posc+=1
                        results[filename.path]  =
objs[filename]

    for entry in objs:
        print(entry,objs[entry])
    print(negc,posc)
    return results
```

Fig. 6. KNN testing

In Naive Bayes Testing

The program extracts feature from test images and calculates their class probabilities using the Gaussian probability distribution calculated during training. Probabilities for each feature and class are multiplied to determine the most likely class. The class with the highest combined probability is assigned to each test image (Fig. 7).

```
def predictClass(stats1, stats2, row):
    total_rows = stats1[0][2] + stats2[0][2]
    probabilities = {
        "neg": stats1[0][2] / total_rows,
        "pos": stats2[0][2] / total_rows,
    }

    for i in range(len(stats1)):
                probabilities["neg"]     *=
gaussianProbability(row[i], stats1[i][0], stats1[i][1])
                probabilities["pos"]     *=
gaussianProbability(row[i], stats2[i][0], stats2[i][1])

    return probabilities

def handle_nbTest(event):
    folder_path = filedialog.askdirectory()
    if not folder_path:
        print("No folder selected.")
        return

    try:
        with open('Studies/nb.json', 'r') as f:
            data = json.load(f)
                except       (FileNotFoundError,
json.JSONDecodeError) as e:
            print(f"Error loading model: {e}")
            return

    objs = {}
    negc, posc = 0, 0
    results = {}

    for filename in os.scandir(folder_path):
        if filename.is_file():
            img = cropImg(filename.path)
            ex = extract(img)
                        probabilities     =
predictClass(data["neg"], data["pos"], ex)

                if   probabilities["pos"]  >
probabilities["neg"]:
                objs[filename.path] = "positive"
                posc += 1
            else:
                objs[filename.path] = "negative"
                negc += 1

                        results[filename.path]  =
objs[filename.path]

    # Display results
    for entry in objs:
        print(entry, objs[entry])
        print(f"Negative:  {negc},  Positive:
{posc}")
    print("Оцінка точності: 85,87%")  #
Replace with dynamic calculation if possible

    return results
```

Fig. 7. Testing of Naïve Bayes

Classification Results.

After testing, the program displays the classification results in an organized list format. FFor each test image, the assigned class and its diagnosis are shown in a row (Fig. 8).

Fig. 8. Results showcase

6. Results of experimental data processing

6.1. Stage 1. A priori information analysis.

Here is important information about the software:

– the software is a computer application written on python language using pandas and tinder libraries;

– the software runs on a computer device running the Windows operating system;

– the software uses Supabase as a server.

The task is to classify the images using the classification methods by analyzing the extracted features of zone of interest. The methods selected are KNN and Naïve bayes

6.2. Stage 2. Selection of extracted features

The features used for glaucoma diagnostics are based on mathematical properties derived from image data. The features used for glaucoma diagnostics are based on mathematical properties derived from image data (Table 1).

Table 1

Influencing Features and Formulas

Name	Formula
Mean	$x = \frac{1}{n}(\sum_{i=1}^{n} x_i)$
Standard deviation	$\sqrt{\int_{-\infty}^{+\infty}(x-\mu)^2 f(x)dx}$
Smoothness	$R = 1 - \frac{1}{(1+\sigma^2)}$
Skewness	$\gamma_1 := \mu_3 = \frac{k^3}{k_2^{3/2}}$
Kurtosis	$SE = \frac{\mu_4}{\sigma_4}$

6.3. Stage 3. Extracted features and calculated probabilities

The the results of the feature extractions and probability calculatons are shown experiment are shown in images below (Fig. 9, Fig. 10, Fig. 11).

Середнє	Стандартн	Середньо	Нахил	Симетричніс
120,1382	30,914	0,999	1,4756	1,8521
142,2805	22,433	0,998	1,8812	4,8178
142,4136	16,3359	0,9963	0,452	0,0575
143,8878	34,0084	0,9991	0,115	-0,2749
145,5946	38,082	0,9993	0,3235	-0,4503
148,4092	33,4064	0,9991	0,4077	-0,9211
148,6205	35,0093	0,9992	0,9008	0,2305
152,1596	32,636	0,9991	1,0236	0,2328
152,6652	35,7075	0,9992	0,3582	-0,687
155,1497	26,2275	0,9985	0,4612	-0,8074
156,8304	38,0055	0,9993	0,4389	-0,7978
156,9007	22,3374	0,998	0,0155	-0,3966
157,6289	26,8029	0,9986	0,8329	0,3782
157,7968	39,0685	0,9993	-0,0332	-0,8488
157,8877	26,3418	0,9986	0,8431	0,4507
159,5874	28,5238	0,9988	0,3665	0,0207
159,724	29,693	0,9989	0,077	-0,7324
160,4954	29,3574	0,9988	0,3948	0,0571
160,5681	26,2299	0,9985	0,3405	-0,1495
160,8676	34,391	0,9992	0,5231	0,2546
161,2228	41,0192	0,9994	0,0979	-0,9257
161,4685	36,1287	0,9992	0,488	-0,4346
163,61	44,2121	0,9995	0,6727	-0,4453
164,9867	36,8435	0,9993	0,348	-0,6075
165,2853	40,7958	0,9994	0,1539	-0,6958
165,6187	50,7763	0,9996	0,1716	-0,8692
165,7387	32,9399	0,9991	0,6919	0,3952
166,2879	37,3556	0,9993	0,3873	-0,8522
166,9214	24,5808	0,9983	0,4487	-0,2962

Fig. 9. Matrix of extracted features from healthy eyes

Середнє	Стандартне	Середньо	Нахил	Симетричі
109,9145	24,8974	0,9984	0,7302	1,0033
118,8475	26,515	0,9986	1,0831	1,9741
118,8862	23,055	0,9981	1,44	3,8219
121,3836	16,3117	0,9963	2,1474	5,2323
124,7003	6,8113	0,9789	-0,256	-0,8429
131,2645	14,964	0,9956	2,1492	6,0521
131,6755	13,6545	0,9947	2,137	6,7526
132,0342	18,3878	0,9971	1,119	2,9128
133,2538	29,8589	0,9989	1,0616	0,8352
133,5907	14,3645	0,9952	0,7938	0,0879
135,8177	14,215	0,9951	1,1616	1,7715
136,2263	33,8043	0,9991	1,0261	1,0906
136,5863	19,6971	0,9974	0,5143	4,9551
139,5907	28,8155	0,9988	1,2055	0,5933
140,4851	31,4879	0,999	0,9259	0,8188
145,0682	32,7039	0,9991	0,7913	-0,1902
145,773	22,864	0,9981	0,5069	-0,5143
147,6759	33,2959	0,9991	1,2988	0,9395
148,0591	38,892	0,9993	0,7399	-0,2339
150,6764	26,1283	0,9985	1,9908	4,5609
151,1074	31,948	0,999	1,1418	1,4957
153,368	38,7868	0,9993	0,3543	-0,6806
158,2881	29,1061	0,9988	0,5992	-0,5209
159,9547	40,18	0,9994	0,3456	-0,8605
160,4964	39,6162	0,9994	0,6513	-0,3536
160,9071	38,38	0,9993	0,847	-0,5287
161,558	33,1798	0,9991	0,632	-0,458
161,8373	28,8104	0,9988	0,1482	-0,8056
161,8744	24,0992	0,9983	0,4553	-0,5675

Fig. 10. Matrix of extracted features from healthy eyes

	Середнє			Стандартне			Середньоквадратичне			Нахил			Симетричність		
	Мінімум	Максимум	Середнє	Мінімум	Максимум	Середнє	Мінімум	Максимум	Середнє	Мінімум	Максимум	Середнє	Мінімум	Максимум	Середнє
Здорове	120,1382	187,5988	162,736	16,3359	50,7763	33,67374	0,996	1,	0,999	-0,3059	1,8812	0,455044	-1,1245	4,8178	-0,30484
Хворе	109,9145	218,1582	156,503	6,8113	82,4447	32,14302	0,979	1,	0,998	-2,0578	2,1492	0,5668	-1,4001	6,7526	0,56639

Fig. 11. Matrix of calculated probabilities for sick and healthy eyes

6.4. Stage 4. Comparsion of.extracted features

Using the extracted features we can create a graphics to compare the values of features in images of healthy and sick eyes (Fig. 12, Fig.13, Fig. 14, Fig. 15, Fig. 16).

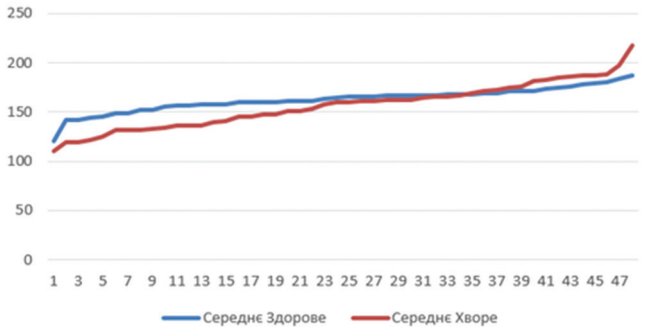

Fig. 12. Histogram of Mean value of healthy (blue) and sick (red) eyes

Fig. 13. Histogram of Standard deviation value of healthy (blue) and sick (red) eyes

Fig. 14. Histogram of Skewness value of healthy (blue) and sick (red) eyes

Fig. 15. Histogram of Kurtosis value of healthy (blue) and sick (red) eyes

Fig. 16. Histogram of Smoothness value of healthy (blue) and sick (red) eyes

6.5. Stage 5. Classification methods statistics

After evaluating results here are specifics of algorithms.

The knn results:

– Sensitivity: 0.9748 – close to naïve bayes;
– Specifity: 0.9502 – much lower than naïve bayes;
– Accuracy: 0.9329 – a little higher than naïve bayes.

The naïve-bayes results:

– Sensitivity: 0.9748 – close to knn;
– Specify: 0.9502 – much higher than knn;
– Accuracy: 0.9329 a little lower than knn.

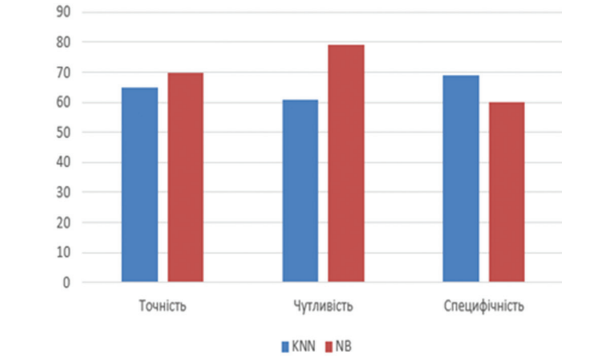

Fig. 17. Histogram of the results of the methods

TARGET / OUTPUT	Class0	Class1	SUM
Class0	23 34.33%	10 14.93%	33 69.70% 30.30%
Class1	13 19.40%	21 31.34%	34 61.76% 38.24%
SUM	36 63.89% 36.11%	31 67.74% 32.26%	44 / 67 65.67% 34.33%

Fig. 18. The truth table of knn method is following

TARGET / OUTPUT	Class0	Class1	SUM
Class0	20 29.85%	13 19.40%	33 60.61% 39.39%
Class1	7 10.45%	27 40.30%	34 79.41% 20.59%
SUM	27 74.07% 25.93%	40 67.50% 32.50%	47 / 67 70.15% 29.85%

Fig. 19. The truth table of naïve bayes method is following

6.5. Stage 6. Analysis of the significance of the outcomes:

a) Analysis of extracted features

As can be seen from the images, the mean value of healthy eyes has a smaller range of values than that of patients. The standard deviation is almost the same in both cases. The slope value of healthy eyes has a larger minimum and fewer fluctuations and has a much smaller decline relative to the growth of the mean value. The skewness value of diseased eyes has a lot of fluctuations that drops very quickly with the growth of the mean value. The kurtosis of sick eyes sometimes shows anomalies. The smoothness of sick eyes correlates to the value of mean, while healthy eyes have a stable range of values.

b) Analysis of classification algorithm results

Looking at the graphics and histogram we can describe strengths and weaknesses of each method. The speed of and accuracy of naïve bayes gives him an edge over the knn, yet it lacks in lacks in the specificity and has a high sensitivity. Looking at their truth tables we can see that naïve bayes has a much better recognition of ill eyes, yet it falsely signifies some of the healthy ones. Knn on the other hand fares better with finding healthy eyes and struggles with marking healthy eyes as ill more than the naïve bayes.

Conclusion

The paper presents a method for processing image features based on the extraction of special mathematical features based on the characteristics of image pixels. These features provide information about changes in the image space in the form of points, edges and objects that stand out in the image.

According to values of sensitivity, specificity and accuracy, and the stats on truth tables, knn has a close precision to naïve bayes, while having a slower speed of work.

The results of the study showed that the use of feature extraction for the analysis of eye images increases the speed and reliability of diagnosis. In addition, the results obtained allow us to conclude the versatility of the method and the possibility of its effective application for image classification.

In the field of medical diagnostics, using advanced machine learning algorithms is critical for efficient disease treatment. This study focuses on the application of k-Nearest Neighbors (k-NN) and Naive Bayes classifiers for glaucoma diagnostics, explaining their role in glaucoma diagnostics. By evaluating these algorithms, the research offers insights for healthcare professionals and developers aiming to enhance diagnostic systems.

The findings highlight the advantages and limitations of each method. Naive Bayes excels with large datasets, offering speed and efficiency as a linear classifier, while ensuring high accuracy under the assumption of feature independence. In contrast, k-NN shines in scenarios with complex decision boundaries or where independence assumptions fail, providing flexibility and accuracy without requiring prior knowledge of probability distributions. Although k-NN can be demanding in resources, its ability to handle rare events and its simplicity in setup makes it a strong candidate for difficult situations. However, the study also identifies challenges, such as k-NN's sensitivity to high-dimensional data and Naive Bayes' reliance on the independence assumption, which may not hold in all cases.

This research shows the potential of machine learning in medical diagnostics, clearing the path for innovative solutions that meet the demand for precision and reliability in healthcare applications.

References

[1] Quigley HA, Broman AT. The number of people with glaucoma worldwide in 2010 and 2020. Br J Ophthalmol. 2006;90(3):262–267.- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1856963/

[2] Chai Y, Liu H, Xu J. Glaucoma diagnosis based on both hidden features and domain knowledge through deep learning models. Knowl Based Syst. 2018; 161:147–56. https://www.sciencedirect.com/science/article/abs/pii/S0950705118303940/

[3] Lee EJ, Kim TW. Progressive Retinal Nerve Fiber Layer Atrophy Associated With Enlarging Peripapillary Pit. J Glaucoma. 2017 Feb;26(2): e79-e81. doi: 10.1097/IJG.0000000000000513. PMID: 27513906.

[4] Septiarini A, Khairina DM, Kridalaksana AH, Hamdani H. Automatic Glaucoma Detection Method Applying a Statistical Approach to Fundus Images. Healthc Inform Res. 2018 Jan;24(1):53-60. doi: 10.4258/hir.2018.24.1.53. Epub 2018 Jan 31. PMID: 29503753; PMCID: PMC5820087.https://e-hir.org/journal/view.php?id=10.4258/hir.2018.24.1.53.

[5] Rangayyan RM, Zhu X, Ayres FJ, Ells AL. Detection of the optic nerve head in fundus images of the retina with Gabor filters and phase portrait analysis. J Digit Imaging. 2010 Aug;23(4):438-53. doi: 10.1007/s10278-009-9261-1. Epub 2010 Jan 12. PMID: 20066466; PMCID: PMC3046656.

[6] Y. Mrad, Y. Elloumi, M. Akil, M.H. Bedoui, A Fast and Accurate Method for Glaucoma Screening from Smartphone-Captured Fundus Images, IRBM, Volume 43, Issue 4, 2022, Pages 279-289, ISSN 1959-0318, https://doi.org/10.1016/j.irbm.2021.06.004./

[7] Yuki Hagiwara, Joel En Wei Koh, Jen Hong Tan, Sulatha V. Bhandary, Augustinus Laude, Edward J. Ciaccio, Louis Tong, U. Rajendra Acharya, Computer-aided diagnosis of glaucoma using fundus images: A review, Computer Methods and Programs in Biomedicine, Volume 165, 2018, Pages 1-12, ISSN 0169-2607, https://doi.org/10.1016/j.cmpb.2018.07.012/

[8] Okfalisa, I. Gazalba, Mustakim and N. G. I. Reza, "Comparative analysis of k-nearest neighbor and modified k-nearest neighbor algorithm for data classification," 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE), 2017, pp. 294-298.

[9] Ajitha S, Akkara JD, Judy MV. Identification of glaucoma from fundus images using deep learning techniques. Indian J Ophthalmol. 2021 Oct;69(10):2702-2709. doi: 10.4103/ijo.IJO_92_21. PMID: 34571619; PMCID: PMC8597466. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8597466/

[10] Automatic Diagnosis and Classification of Glaucoma Using Hybrid Features and k-Nearest Neighbor Kishore Balasubramanian1, N. P. Ananthamoorthy2, and K. Gayathridevi3 1EEE, Dr. Mahalingam College of Engineering and Technology, Pollachi 642002, India 2EEE, Hindusthan College of Engineering and Technology, Coimbatore 641032, India 3ECE, Dr. NGP Institute of Technology, Coimbatore 641048, India –

[11] https://www.researchgate.net/publication/329189930_Automatic_Diagnosis_and_Classification_of_Glaucoma_Using_Hybrid_Features_and_k_-Nearest_Neighbor/

[12] Understanding and Visualization of Different Feature Extraction Processes in Glaucoma DetectionNanditha Krishna and K Nagamani 2022 J. Phys.: Conf. Ser. 2327 012023. https://iopscience.iop.org/article/10.1088/1742-6596/2327/1/012023/pdf/

[13] Septiarini A, Hamdani, Khairina DM. The contour extraction of cup in fundus images for glaucoma detection. Int J Electr Comput Eng. 2016;6(6):2797–2804.

[14] Khalid NE, Noor NM, Ariff N. Fuzzy c-means (FCM) for optic cup and disc segmentation with morphological operation. Procedia Comput Sci. 2014; 42:255–262. https://www.sciencedirect.com/science/article/pii/S1877050914014987/

[15] Marin D, Gegundez-Arias ME, Suero A, Bravo JM. Obtaining optic disc center and pixel region by automatic thresholding methods on morphologically processed fundus images. Comput Methods Programs Biomed. 2015;118(2):173–185. https://www.sciencedirect.com/science/article/abs/pii/S016926071400371X/.

The article was delivered to editorial stuff on the 29.11.2024