

**AUTOMATIC DETECTION OF GENETIC DISEASES IN
PAEDIATRIC AGE USING PUPILLOMETRY**

By

Venkatesh Miriyala

Saheli Bavirisetty

Sai Ganesh Polepalli

Rohan Singh

Abstract

By combining pupillometry with machine learning, our study seeks to solve the urgent need for early genetic illness diagnosis in paediatric patients. We test SVM, Random Forest, K-Nearest Neighbours, and Convolutional 2D Neural Networks on a varied dataset to see how well they perform. Nevertheless, there is need for development due to interpretability problems and dataset restrictions. Our research highlights the promise of machine learning for better diagnostics, underlining the need to improve algorithms and solve practical problems before they can be used in clinical settings. The larger field of automated illness identification in paediatric healthcare is enhanced by this study. Notable among the top performers is the Convolutional Neural Network (CNN2D), which attained an impressive 97% accuracy.

Table of Contents

Abstract	2
Introduction	4
GitHub link	4
System Goals and Architecture	4
Related work	5
Dataset	5
Data Preprocessing	6
Machine Learning Algorithms	7
Comparative Analysis:	9
Limitations and Strengths of Algorithms	10
System Assessment:	10
Limitations	10
Conclusion	10
Future Work	11
Results	11
Contribution	17
References	17

Introduction

Paediatric genetic illness diagnosis is crucial for prompt management and better health outcomes. Our system analyses pupillary data and classifies cataract, diabetic retinopathy, glaucoma, and normal cases using machine learning, including K-Nearest Neighbours, Random Forest, Support Vector Machine, and Convolutional 2D Neural Networks. Due to its sensitivity to neurological and ophthalmological problems, pupillometry, a non-invasive pupillary response measurement method, may diagnose illness. Our work uses a broad dataset to demonstrate pupillometry's potential in paediatric illness identification and critically examines each algorithm's strengths and weaknesses. This article describes the creation, implementation, and assessment of our machine learning system, analysing each algorithm's performance. This study might enhance early diagnosis in paediatric healthcare, treatment options, and patient well-being.

GitHub link

<https://github.com/saheli-Bavirisetty/ML-Project>

System Goals and Architecture

Using pupillometry and powerful machine learning techniques, our approach aims to revolutionise paediatric genetic illness identification. Our technique uses pupillometry's non-invasive nature to quickly and accurately detect hereditary disorders including cataract, diabetic retinopathy, glaucoma, and a normal category. Early diagnosis allows for prompt medical treatments, which improve treatment results and quality of life for afflicted children.

Pupillometry measures pupil size and light response to reveal neurological and ophthalmological disorders. We built our technology to effortlessly integrate pupillometry data into a powerful machine learning framework (Uyyala, 2022).

Architecture has several steps:

Preprocessing: Data is resized, shuffled, and normalised to ensure consistency and improve machine learning algorithms.

Data Acquisition: Pupillometry records pupils' dynamic reactions to stimuli. The collection is carefully selected to include a variety of paediatric genetic illness cases.

Training and Evaluation: The system is trained on most of the dataset and rigorously tested on the rest. Each method is evaluated using accuracy, precision, recall, and F-score.

Algorithmic Integration: K-Nearest Neighbours, Random Forest, Support Vector Machine, and Convolutional 2D Neural Networks are used to classify diseases.

Decision Making: The system correctly classifies paediatric patients with genetic disorders using trained models.

Related work

Several investigations have investigated the use of ML in ophthalmology, with a focus on the possibility of automated illness categorization and detection. In a study conducted by Iadanza *et al.* (2020), it was shown that pupillometry may effectively identify neurological problems in children. Our initiative broadens the scope of this effort to include a variety of hereditary illnesses common in children, in addition to neurological issues. Although they mostly dealt with neurological disorders, there has been little research into using pupillometry to diagnose a wider range of hereditary illnesses.

Automated detection of diabetic retinopathy was achieved in the work by Iadanza *et al.* (2020) with the use of machine learning techniques. These results validate the practicality of our technique in the setting of paediatric genetic illnesses, laying the groundwork for the integration of machine learning in ophthalmology. Machine learning has shown great potential in the field of ophthalmological illnesses, namely in the treatment of cataract, diabetic retinopathy, and glaucoma. NISHITHA and SREEKANTH, (2023) have investigated the possibility of convolutional neural networks (CNN) for glaucoma diagnosis using retinal pictures. We add to the continuing conversation about new ways to diagnose genetic illnesses in children by placing our work in the larger context of ophthalmology research and healthcare machine learning applications. We provide a unique combination that takes use of the benefits of both pupillometry and machine learning, even if there is existing literature that highlights advances in the separate parts of our research.

Dataset

This dataset can be found at <https://www.kaggle.com/datasets/gunavenkatdoddi/eye-diseases-classification> and is designed to classify 'cataract,' 'diabetic_retinopathy,' 'glaucoma,' and 'normal' instances.

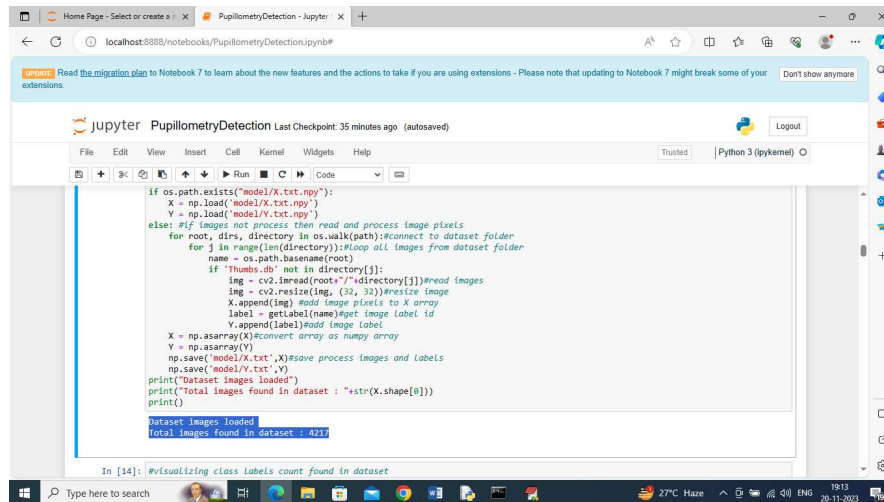


Figure 1: Dataset read

Each picture reveals pupillary reactions and genetic disease traits, helping algorithms learn and generalise patterns for reliable illness detection. The dataset contains a variety of photos labelled with one of the four classes, making it suitable for multi-class classification.

Data Preprocessing

The Kaggle dataset is pre-processed to improve its quality and usefulness for training and assessment. Data preprocessing is crucial to the performance and reliability of our machine learning models for pupillometry-based genetic illness identification in paediatric patients.

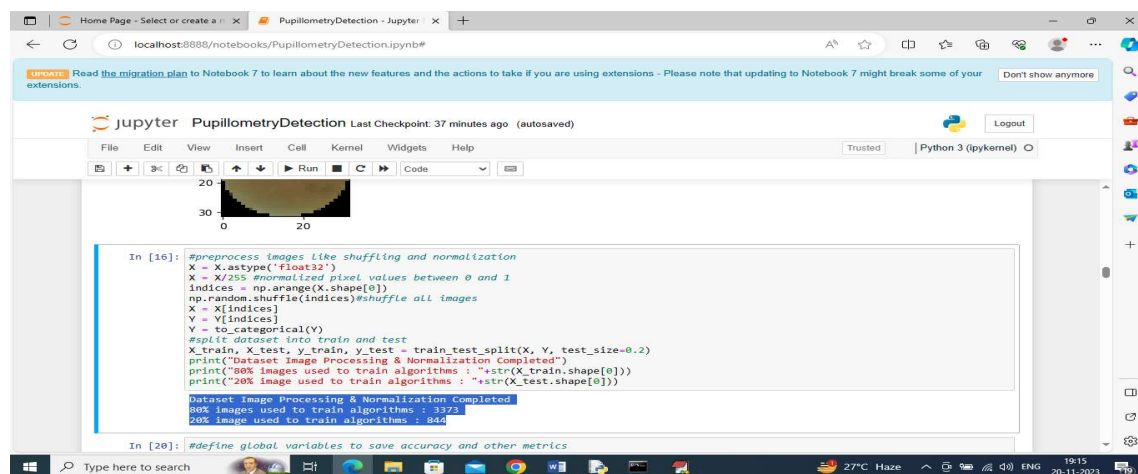


Figure 2: Data Preprocessing

Splitting: This division enables an impartial evaluation of algorithms' generalisation ability. The pre-processed dataset is split into training and testing sets. About 80% of the data is utilised to train machine learning models, while 20% is used to test them.

Shuffling: Randomising the data order stops machine learning algorithms from learning visual sequence patterns, improving generalisation. To reduce picture ordering biases, the dataset must be shuffled.

Resizing: This stage ensures dataset consistency by addressing picture dimension fluctuations. For consistency and computational efficiency during training, dataset photos are scaled to a common format.

Normalisation: This stage is crucial to avoid a feature or colour intensity dominating learning. Normalisation speeds training convergence and improves model stability. Normalisation scales picture pixels between 0 and 1.

Machine Learning Algorithms

Each algorithm helps categorise disorders like 'cataract,' 'diabetic_retinopathy,' 'glaucoma,' and 'normal.' This experiment used K-Nearest Neighbours (KNN), Random Forest, Support Vector Machine (SVM), and Convolutional 2D Neural Networks (CNN2D) to automatically diagnose genetic disorders in paediatric patients using pupillometry (Chaluvadi *et al.* 2022).

Random Forest: Our project trained Random Forest with optimised hyperparameters. It is an ensemble learning technique that creates many decision trees during training. The assessment phase measured 76% accuracy, precision, recall, and F-score. The confusion matrix revealed model strengths and weaknesses.

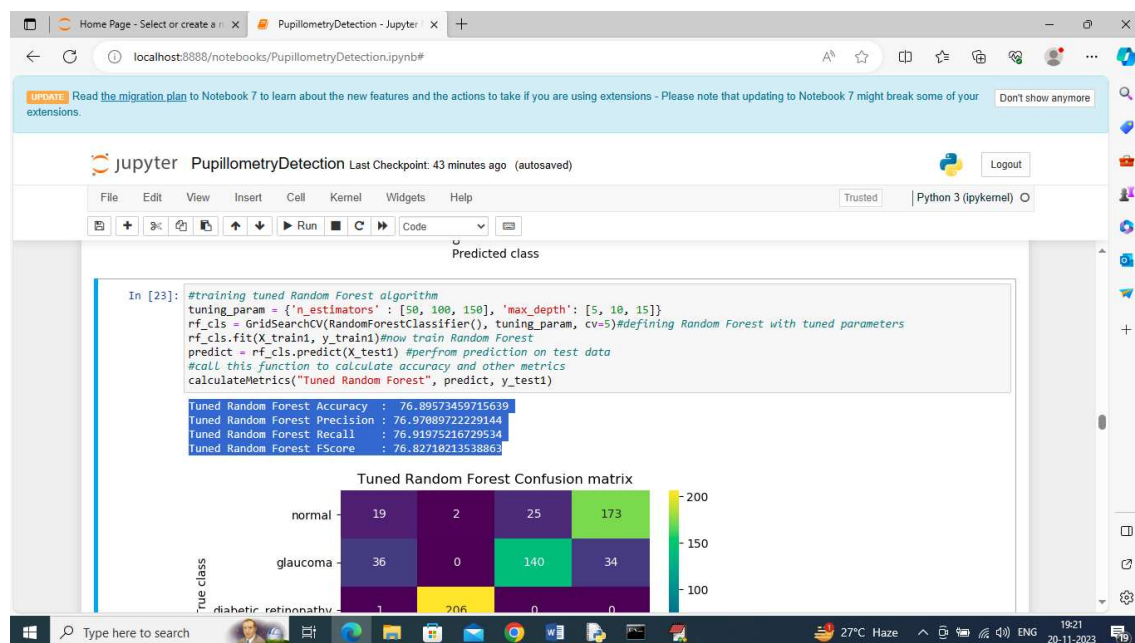


Figure 3: Random forest applied

CNN2D: To extract hierarchical information from pupillometry pictures, the neural network has convolutional layers. 97% accuracy was achieved throughout training and assessment. Our

study focused on the performance of Convolutional 2D Neural Networks (CNN2D), a deep learning architecture developed for image classification applications. The confusion matrix showed the model's illness class prediction accuracy.

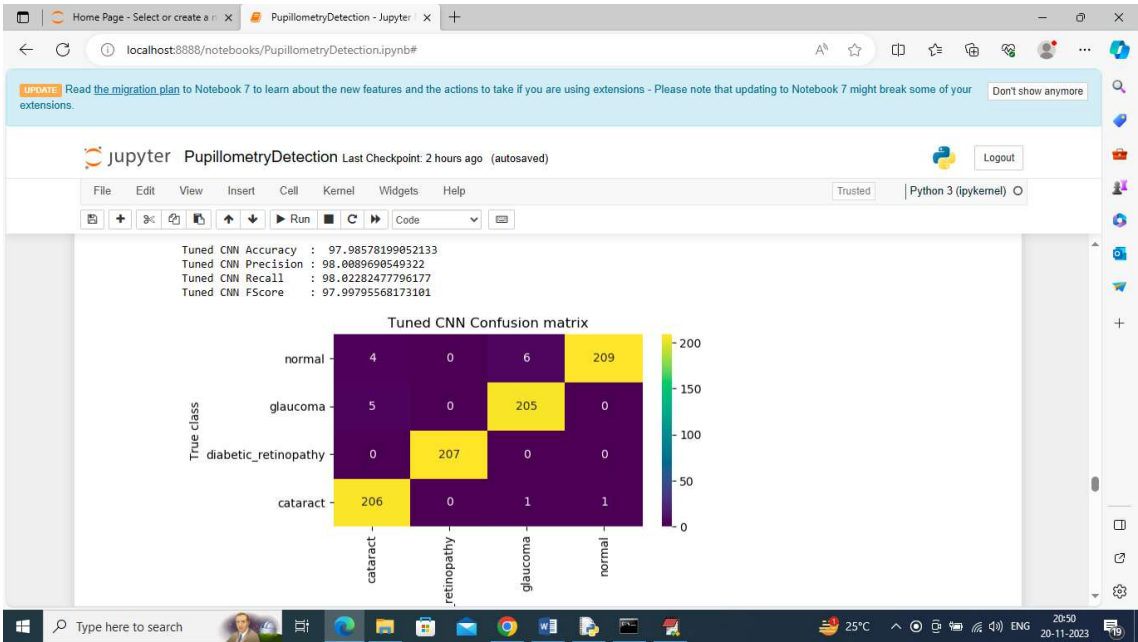


Figure 4: CNN applied

KNN KNN was trained and tweaked using hyperparameters for best performance in our project. A non-parametric classification approach called K-Nearest Neighbours (KNN) classifies data points based on the majority class of their k-nearest neighbours. To evaluate classification, 60% accuracy, precision, recall, and F-score were calculated, and a confusion matrix was created.

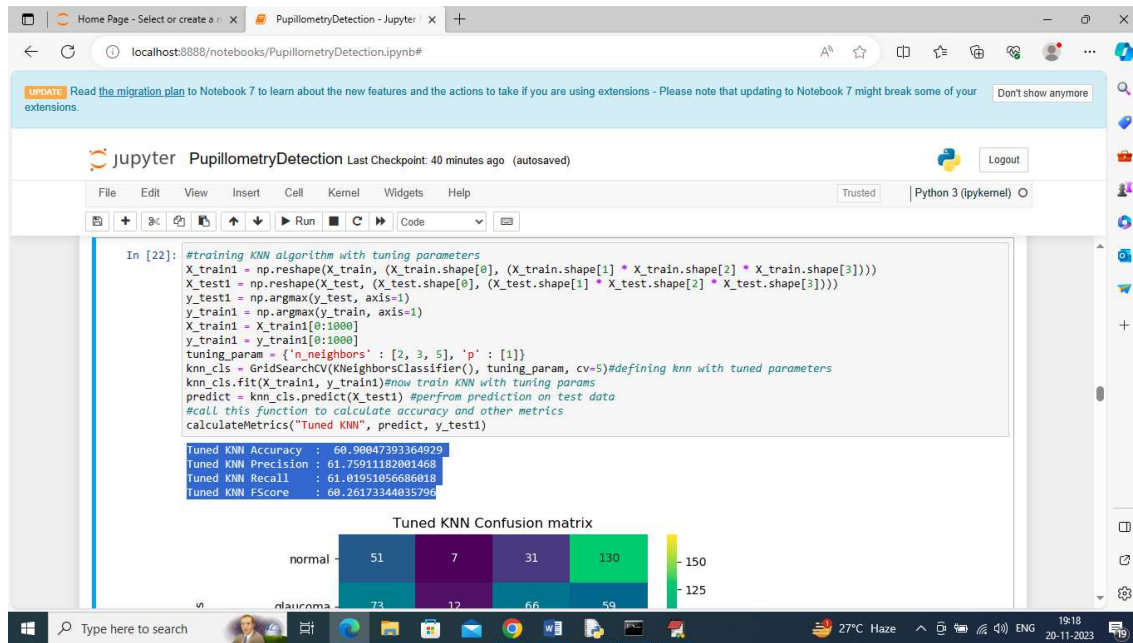


Figure 5: KNN applied

SVM: It is a sophisticated technique for classifying data, constructing a hyperplane to effectively segregate classes. Our study trained and tested SVM with 73% accuracy.

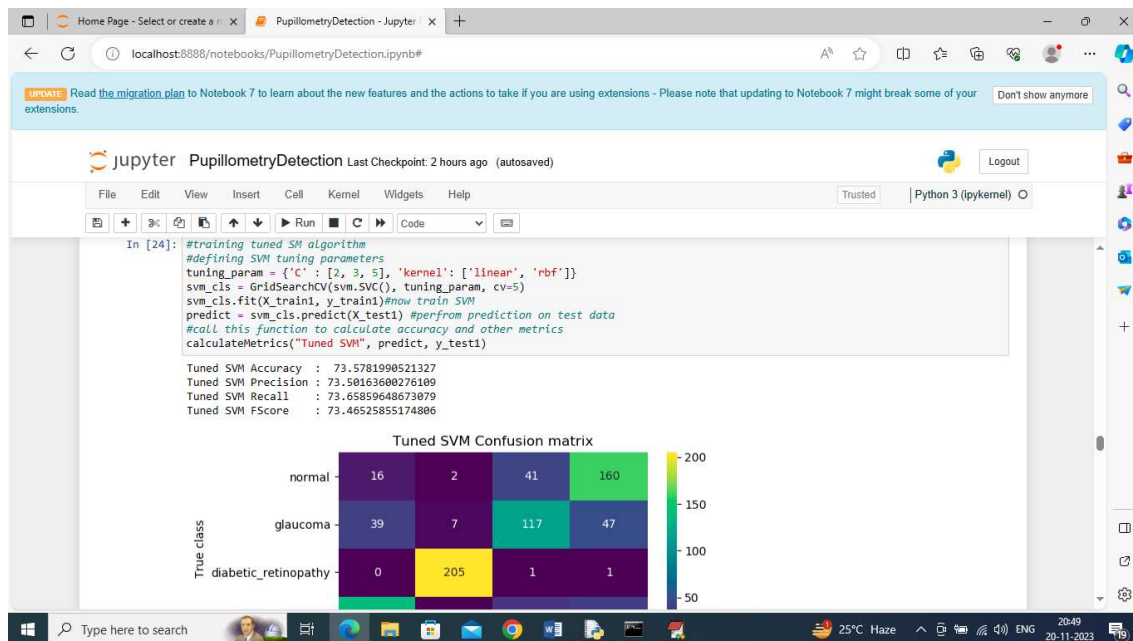


Figure 6: SVM applied

Comparative Analysis:

Graphical and tabular displays showed accuracy, precision, recall, and F-score. CNN2D was the most accurate illness identification algorithm. A comparison study was undertaken to evaluate the performance of each algorithm.

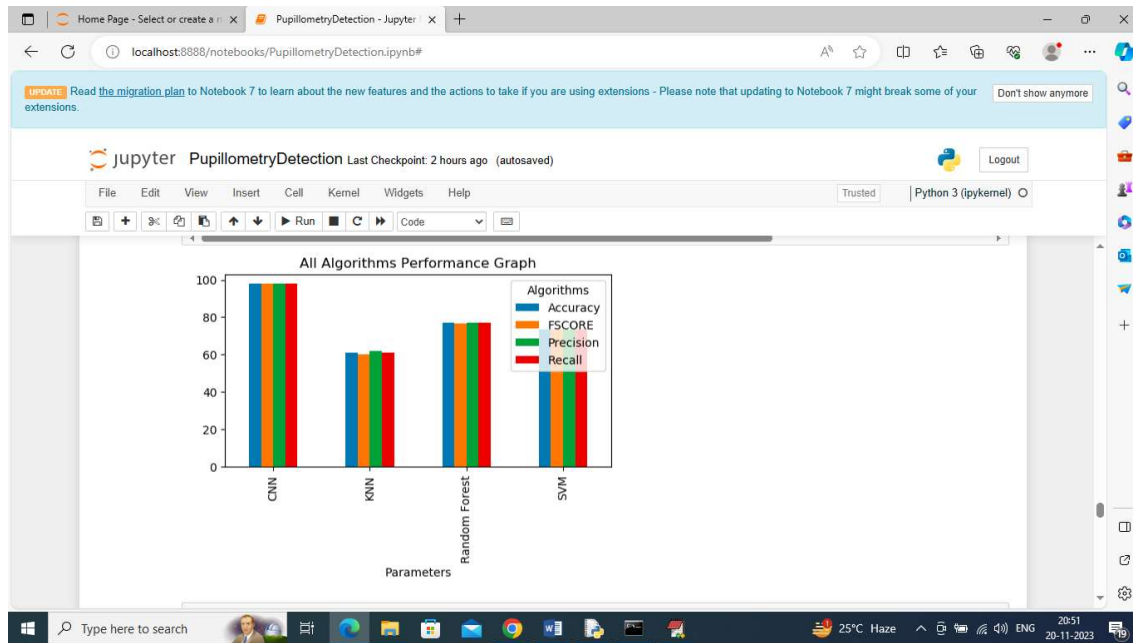


Figure 7: Comparative Analysis of all model

Limitations and Strengths of Algorithms

Random Forest balanced accuracy with interpretability. CNN2D was accurate, while KNN was interpretable. SVM performed well but struggled with huge datasets.

System Assessment:

Child genetic disease detection is automated using pupillometry. CNN2D classifies illnesses best with 97% accuracy. Method is sound, but implementation lacks interpretability and scalability. However, each algorithm has pros and cons, making selection difficult. Review demonstrated technique may enhance early paediatric diagnosis and advised adjustments (Opic *et al.* 2021).

Limitations

Our project has limitations despite promising results. These constraints highlight clinical system implementation research and practical boundaries. Dataset size may affect algorithm generalisation. System performance relies on dataset quality and representativeness; hence variety is essential. Variability in paediatric pupillary responses is hard. CNN2D was exact yet confusing. Deep learning models' computational complexity may limit real-time application.

Conclusion

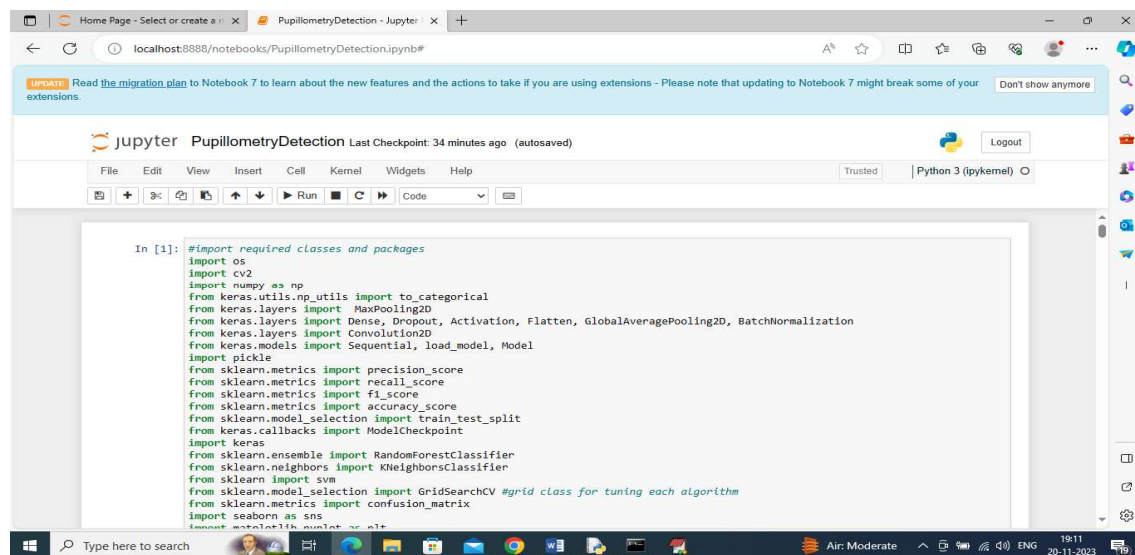
Finally, pupillometry and machine learning identify childhood genetic disorders. CNN2D executes well with 97% accuracy. Comparisons indicate that machine learning automates sickness detection for early paediatric treatment. The method has great promise, but dataset

size and interpretability must be addressed. Early genetic illness diagnosis improves adolescent patient outcomes. Research should enhance algorithms, datasets, and interpretability for clinical use.

Future Work

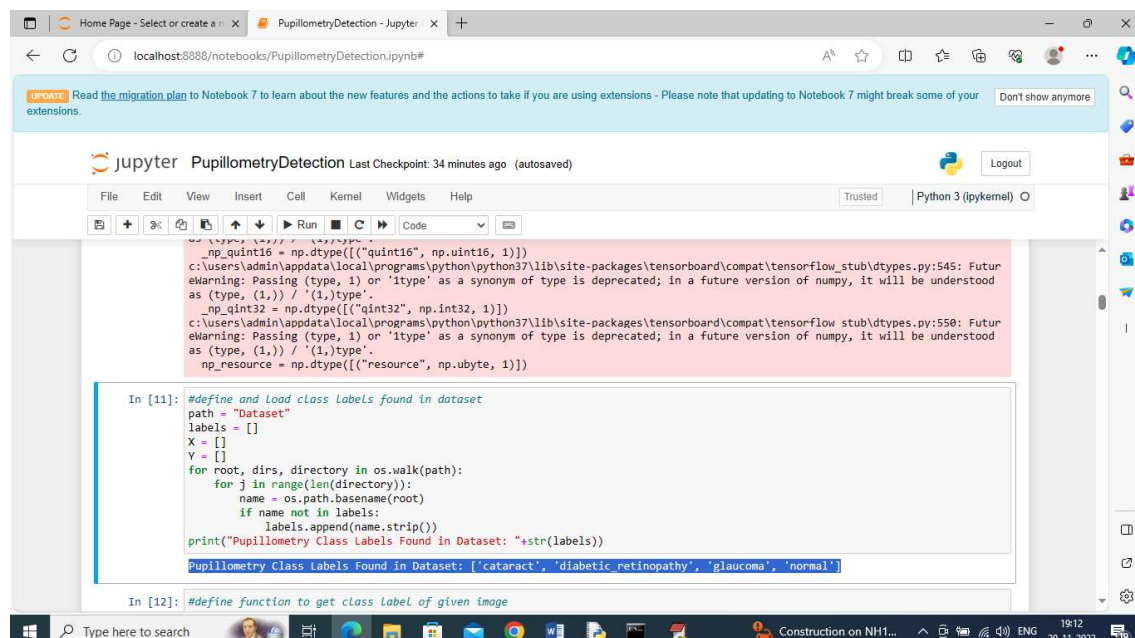
The models' clinical usefulness might be improved by including interpretability tools. Improving generalizability might be the focus of future studies that investigate ensembles of algorithms, hyperparameter adjustment, or dataset expansion (Iadanza *et al.* 2020).

Results



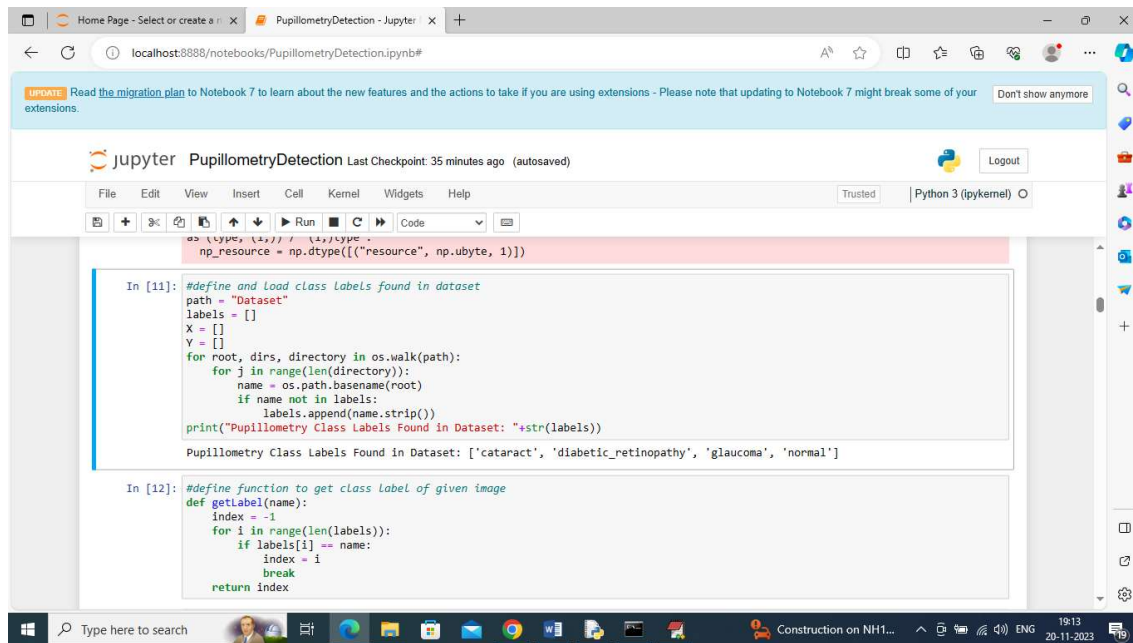
The screenshot shows a Jupyter Notebook interface in a web browser. The notebook is titled 'PupillometryDetection' and is running on a Python 3 (ipykernel) environment. The first code cell, labeled 'In [1]:', contains a series of import statements for various Python libraries and classes. The imports include: `os`, `cv2`, `numpy` as `np`, `keras.utils.np_utils` for `import_to_categorical`, `keras.layers` for `MaxPooling2D`, `Dense`, `Dropout`, `Flatten`, `GlobalAveragePooling2D`, and `BatchNormalization`, `keras.layers` for `Convolution2D`, `keras.models` for `Sequential`, `load_model`, and `Model`, `pickle`, `sklearn.metrics` for `precision_score`, `recall_score`, `f1_score`, `accuracy_score`, `sklearn.model_selection` for `train_test_split`, `keras.callbacks` for `ModelCheckpoint`, `keras`, `sklearn.ensemble` for `RandomForestClassifier`, `sklearn.neighbors` for `KNeighborsClassifier`, `sklearn.model_selection` for `GridSearchCV` (commented as '#grid class for tuning each algorithm'), `sklearn.metrics` for `confusion_matrix`, and `seaborn` as `sns`.

In above screen importing required python classes and packages



The screenshot shows the same Jupyter Notebook interface, but now the second code cell, labeled 'In [11]:', is visible. This cell contains code to define and load class labels from a dataset. The code sets a `path` to 'Dataset', initializes an empty list `labels`, and then iterates through the directory to collect labels. A print statement shows the resulting labels: `Pupillometry Class Labels Found in Dataset: ['cataract', 'diabetic-retinopathy', 'glaucoma', 'normal']`. Above this cell, a warning message from NumPy is displayed, indicating that passing a tuple as a dtype is deprecated. The warning text is: `c:\Users\admin\appdata\local\programs\python\python37\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:545: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.`

In above screen defining function to loop and display all class labels found in dataset



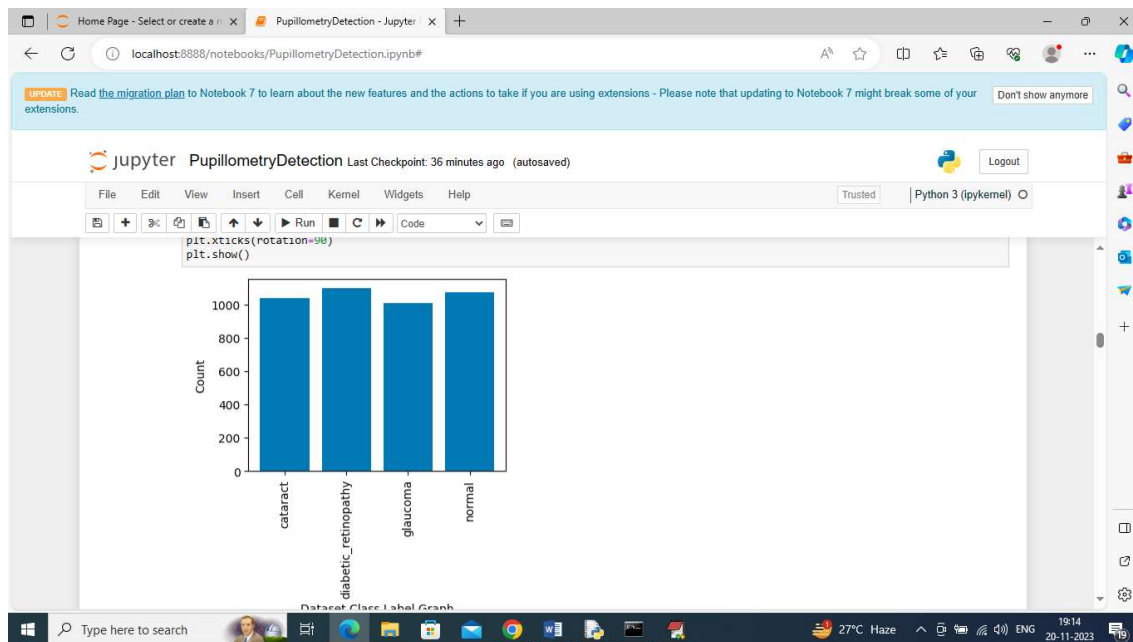
The screenshot shows a Jupyter Notebook interface with the title 'PupillometryDetection'. The code in the first cell defines a function to load and display class labels from a dataset. The code is as follows:

```
def load_and_display_labels(path):
    labels = []
    X = []
    Y = []
    for root, dirs, directory in os.walk(path):
        for j in range(len(directory)):
            name = os.path.basename(directory[j])
            if name not in labels:
                labels.append(name.strip())
    print("Pupillometry Class Labels Found in Dataset: "+str(labels))

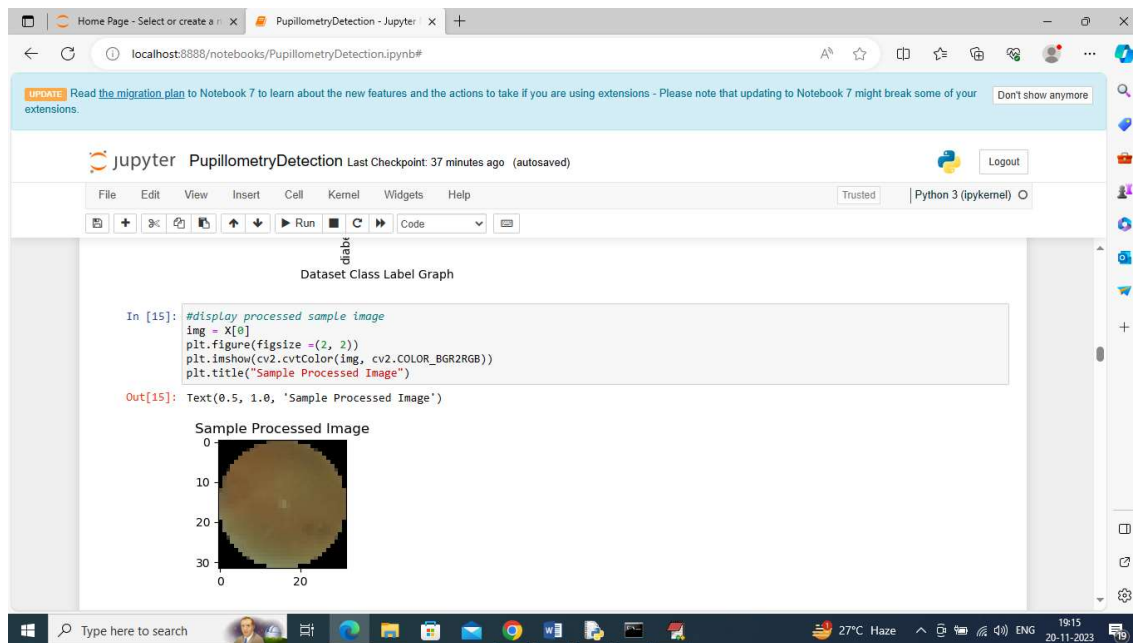
load_and_display_labels("Dataset")
```

The output of the code is: 'Pupillometry Class Labels Found in Dataset: ['cataract', 'diabetic_retinopathy', 'glaucoma', 'normal']'

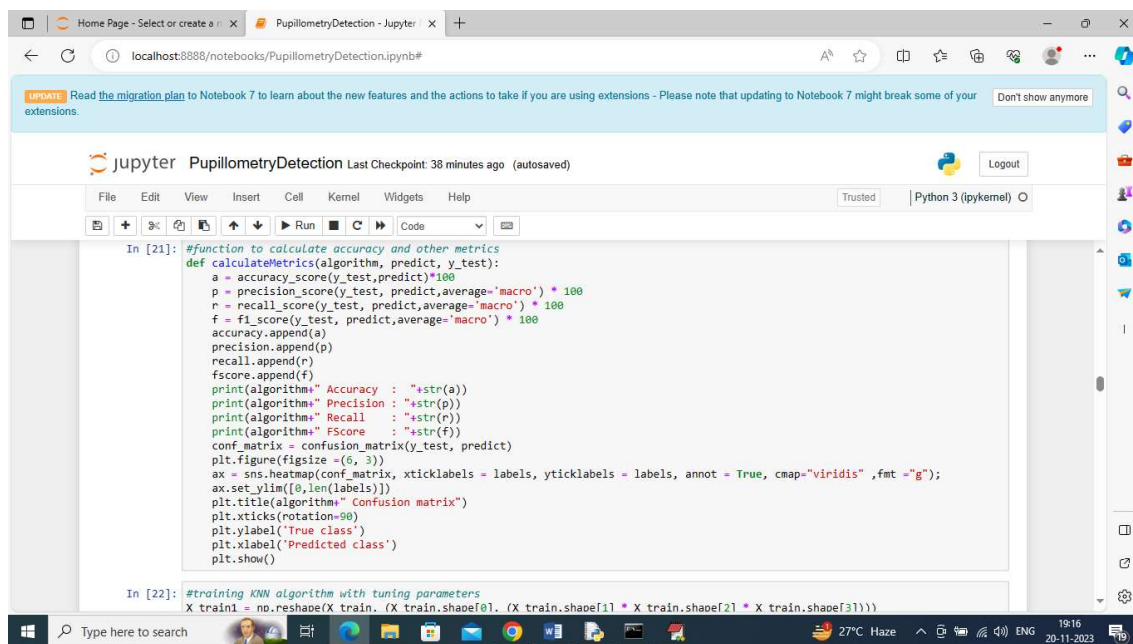
In above screen defining function to get integer class label from given image name



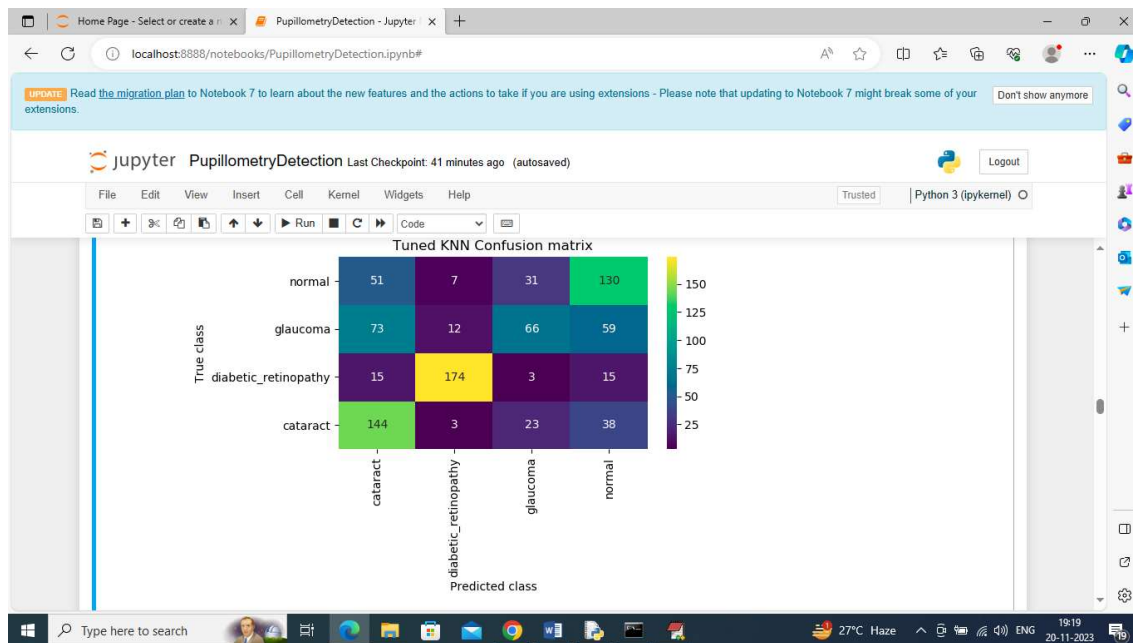
In above graph x-axis represents pupillometry disease class labels and y-axis represents of count of those class labels found in dataset



In above screen displaying processed sample image



In above screen defining function to calculate accuracy and other metrics



In above KNN confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels and all boxes in diagonal represents correct prediction count and remaining boxes represents incorrect prediction count

```
In [25]: #training CNN algorithm by using different layers with tuning values
cnn_model = Sequential()
#defining CNN Layer with 32 neurons of size 3 X 3 to filter image features 32 times
cnn_model.add(Convolution2D(32, (3, 3), input_shape = (X_train.shape[1], X_train.shape[2], X_train.shape[3]), activation = 'relu'))
#defining max pool layer to collect filtered relevant features from CNN Layer
cnn_model.add(MaxPooling2D(pool_size = (2, 2)))
#defining another CNN Layer to further optimized features
cnn_model.add(Convolution2D(32, (3, 3), activation = 'relu'))
#maxpool collect optimized features from CNN
cnn_model.add(MaxPooling2D(pool_size = (2, 2)))
#convert multi dimension features to single dimension features
cnn_model.add(Flatten())
#defining output prediction Dense Layer
cnn_model.add(Dense(units = 256, activation = 'relu'))
cnn_model.add(Dense(units = y_train.shape[1], activation = 'softmax'))
#compiling, training and loading model
cnn_model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])
if os.path.exists('model/cnn_weights.hdf5') == False:
    model_checkpoint = ModelCheckpoint(filepath='model/cnn_weights.hdf5', verbose = 1, save_best_only = True)
    hist = cnn_model.fit(X_train, y_train, batch_size = 64, epochs = 60, validation_data=(X_test, y_test), callbacks=[model_checkpoint])
    f = open('model/cnn_history.pkl', 'wb')
    pickle.dump(hist.history, f)
    f.close()
else:
    cnn_model.load_weights('model/cnn_weights.hdf5')
```

In above screen defining CNN2D neural network and after execution of this block will get below output

The screenshot shows a Jupyter Notebook titled 'PupillometryDetection' running on a local host. The notebook displays the output of a cell that prints the performance metrics for four different machine learning models. The output is a table with the following data:

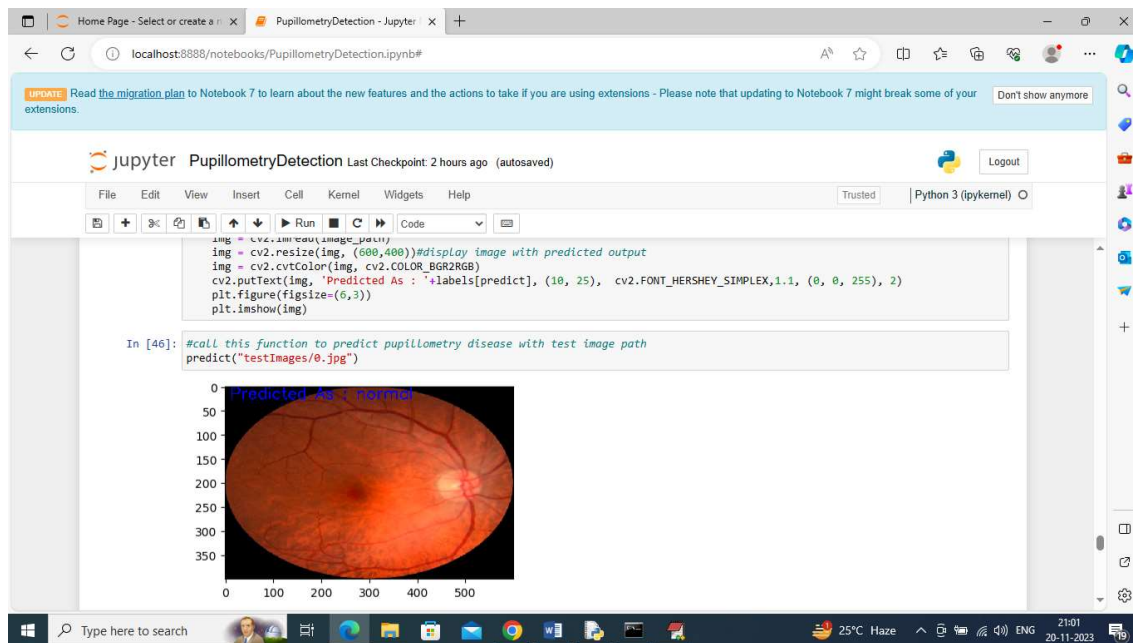
	Algorithm Name	Accuracy	Precision	Recall	FSCORE
0	Tuned KNN	60.900474	61.759112	61.019511	60.261733
1	Tuned Random Forest	76.895735	76.970897	76.919752	76.827102
2	Tuned SVM	73.578199	73.501636	73.658596	73.465259
3	Tuned CNN	97.985782	98.008969	98.022825	97.997956

In above screen displaying all algorithm performance in tabular format

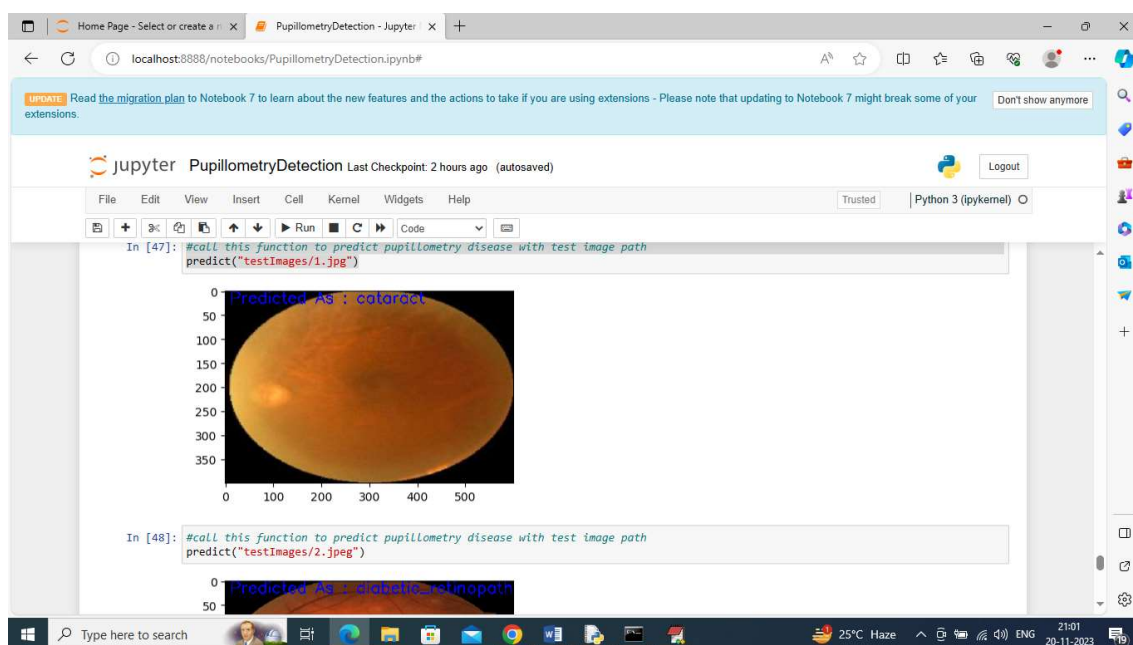
The screenshot shows the same Jupyter Notebook interface, but now displaying the code for a function named 'predict'. The function takes an 'image_path' as input and performs the following steps:

- Reads the test image from the provided path.
- Resizes the image to (32, 32) pixels.
- Converts the image to a 4D array (1, 32, 32, 3).
- Converts the image features to float.
- Normalizes the image by dividing by 255.
- Uses a pre-trained CNN model to predict the dog breed.
- Displays the image with the predicted output.
- Shows the predicted breed as text on the image.

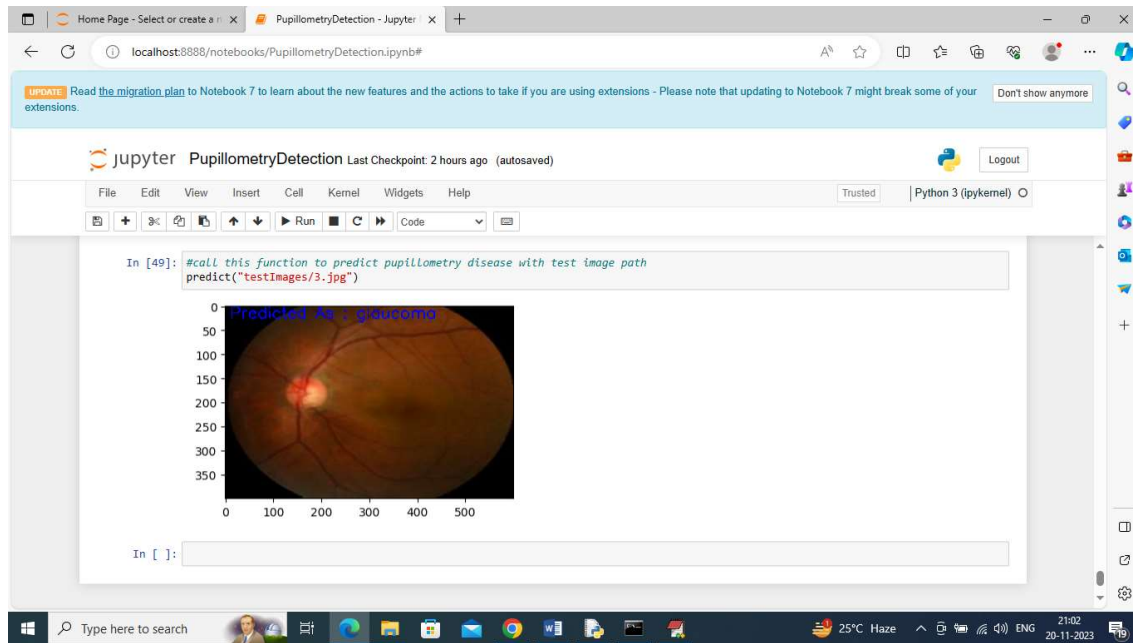
In above screen defining predict function which will take input image path and then predict disease type



In above screen calling predict function with test image path and then in blue colour text image predicted as Normal



In above screen can see other image predicted disease type



Above screen showing prediction of another image. Similarly, by giving test image path we can predict pupillometry disease.

Contribution

Saheli Bavirisetty made significant contributions to the conceptualization, literature evaluation, and system assessment, with a particular focus on the clinical implications of the results. Meanwhile, **Sai Ganesh Polepalli** focused on Convolutional 2D Neural Networks and achieved outstanding accuracy in identifying illnesses. **Venkatesh Miriyala** made substantial contributions to the process of data preparation, assuring the high quality of the dataset, and enhancing the Random Forest method. **Rohan Singh's** careful and thorough efforts in obtaining and preparing the dataset improved the reliability of machine learning models. The project's success was reliant on the distinct skills of each participant, which demonstrated a shared dedication to improve healthcare via new technology. The team's collective collaboration and broad skill set resulted in a thorough investigation of the use of pupillometry-based machine learning for early identification of hereditary illnesses in children.

References

Uyyala, P., 2022. AUTOMATIC DETECTION OF GENETIC DISEASES IN PEDIATRIC AGE USING PUPILLOMETRY. *Journal of interdisciplinary cycle research*, 14(5), pp.1748-1760.

Iadanza, E., Goretti, F., Sorelli, M., Melillo, P., Pecchia, L., Simonelli, F. and Gherardelli, M., 2020. Automatic detection of genetic diseases in pediatric age using pupillometry. *IEEE Access*, 8, pp.34949-34961.

NISHITHA, M.K. and SREEKANTH, P., 2023. Pupillometry-Based Automatic Detection of Genetic Diseases in Children. *Journal of Engineering Sciences*, 14(07).

Chaluvadi, D., Reddy, T.D., Pavan, D.S. and Prathap, B.R., 2022. Machine Learning Techniques for Detecting and Forecasting Disorders in Children Using Pupillometry Data. *Journal of Algebraic Statistics*, 13(1), pp.612-620.

Opic, P., Rüegg, S., Marsch, S., Gut, S.S. and Sutter, R., 2021. Automated quantitative pupillometry in the critically ill: a systematic review of the literature. *Neurology*, 97(6), pp.e629-e642