International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 • Website: www.ijfmr.com • Email: editor@ijfmr.com

Screen Based Vision Acuity Prediction Using Machine Learning and Failure Threshold Analysis

K. Sudheendra[1], R. Thanmayi[2], Mrs. N. Musrat Sulthana[3], Mr. P. Satya Shekar Varma[4]

[1,2]Undergraduate Students, Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Hyderabad, Telangana, India
[3,4]Assistant Professors, Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Hyderabad, Telangana, India

Abstract
Sight or vision is one of the most vital functions that enable an individual to perceive his/her surroundings and interact effectively. One of the most prevalent eye conditions affecting millions of people across the globe is refractive errors such as myopia and hypermetropia. Identifying these eye problems as early as possible is necessary to help prevent further complications. Vision tests done traditionally may need some level of clinical facilities together with other factors. It becomes very difficult to achieve the same through conventional means. This project aims at providing a solution to these problems through the introduction of an online vision test tool that estimates eye power based on visual acuity. The proposed method involves displaying random characters at various sizes using a digital screen and then increasing or decreasing their complexity depending on the performance of a user. Failure thresholds which refer to the smallest font size that can be read by the user are used as the major parameter in estimating eye power. Machine learning models are then used to classify eye power and predict whether the individual has myopia, hypermetropia or any other eye condition. This approach also highlights the ease-of-use of this method since the vision test will not require the use of expensive medical equipment or the constant presence of professional personnel. The implementation of adaptive algorithms ensures the high accuracy of the testing process by personalizing the examination according to each individual's capabilities. Thus, mistakes and redundant tests are minimized. Moreover, machine learning allows the system to constantly refine its predictive capacity by studying testing results. This vision test can be utilized as an efficient tool for preliminary screening among masses of people, particularly in areas where facilities providing eye care services are lacking.

Keywords: Vision acuity, adaptive testing, myopia, hypermetropia, screen-based vision testing, machine learning, Random Forest, failure threshold, vision prediction, graphical user interface.

1. Introduction

Vision loss due to refractive error, which includes myopia, hypermetropia, and astigmatism, is undoubtedly one of the greatest yet entirely avoidable challenges to global health care. According to the

World Health Organization, there are more than 2.2 billion people worldwide suffering from vision impairment, at least half of whom have problems associated with conditions that could be easily prevented or corrected [WHO, 2019]. Unfortunately, detecting these eye diseases early enough before they cause permanent damage is not an easy task, especially in developing nations where ophthalmological care and professionals are scarce.

Standard visual acuity testing requires presenting optotypes, which include such charts as Snellen or LogMAR chart, under certain lighting conditions, from a particular testing distance, and using specific optotypes in a professional environment [14]. While reliable and accurate, this approach is static and prone to optotype memorization, making it impractical for independent self-use via a software solution. The growing number of personal computers and screens can thus be used as an opportunity to move visual testing to a new software environment.

Current methods for digital vision testing include systems that use smartphones, such as Peek Acuity [1], the K-VA logMAR test [6], and V@home [12]. These applications have shown potential correlation with standard Snellen tests. Yet, all these solutions suffer from one key drawback—they calculate visual acuity and do not determine the refractive error directly in diopters, which is the desired result for prescribing corrective lenses. Moreover, most of these applications require certain mobile devices, camera calibration, or light control, thus limiting the scope of application. There is still room for developing an adaptive, platform-independent, desktop solution that can evaluate visual acuity and predict refractive error without any external equipment.

1.1 Research Gap

As pointed out in the literature review section above, there currently does not exist any vision assessment system that integrates both the adaptive character size testing and machine learning-based refractive error prediction into one system. Existing methods either concentrate exclusively on simple visual acuity testing or require Internet connection and clinical assistance from outside sources. Moreover, most of the existing methods do not sufficiently solve the problem of memorization during repetitive testing. To circumvent such difficulties, the system under development incorporates an adaptive character size testing algorithm which ensures minimal memorization and automatic adjustment of the test difficulty according to the responses of the user. Besides, machine learning algorithms are used to transform the results of the adaptive testing process into numerical estimates of refractive errors. The whole system is designed to be portable and stand-alone, and operates using only the resources of the computer itself without requiring an Internet connection.

1.2 Research Contributions

Key contributions of this study consist of designing a new computer adaptive testing (CAT) algorithm for visual acuity test on standard desktop monitors using random character selection technique to avoid memorization during the repetitive tests. Failure threshold based systematic feature extraction algorithm has been also proposed in this study that will convert the output of the adaptive testing into a compact feature vector that can be analyzed and predicted further. In addition, a machine learning algorithm named Random Forest Regression has been designed, trained and validated to quantify the degree of refractive error in terms of diopters by comparing its performance with five other regression algorithms. Moreover, the whole system has been developed into a portable desktop software application that can run completely independent from the internet and any special clinical device. Finally, performance validation experiment of the designed software application has been carried out using 232 subjects data sets.

2. Related Work

Research on digital vision testing has evolved along two principal trajectories: hardware-accelerated smartphone applications that reproduce clinical optotype testing, and software-driven adaptive approaches that employ algorithmic adjustments based on user performance. This section critically reviews representative works from both domains, identifying methodological strengths, limitations, and the research gap that motivates the present contribution.

2.1 Smartphone-Based Visual Acuity Testing

In a clinical trial, Bastawrous et al. [1] devised Peek Acuity, an application for smartphones that shows Tumbling-E optotype and performs visual acuity tests through the forced-choice technique. Validation tests done on standard logMAR charts showed good consistency (ICC>0.90), and therefore the feasibility of handhelds as platforms for testing visual acuity. Nevertheless, the system lacks adaptive capability except for the beginning of the test and requires manual proctoring, which means it is not truly self-administrable. According to Aruljyothi [11], validation tests by Bhaskaran et al. showed the need to calibrate devices because of the difference in the angular subtense.

A comparative study by Bhaskaran et al. [2] was conducted between various VA tests from applications to Snellen's chart, using adults in India. It was concluded that even though applications provide accurate measurements of visual acuity with acceptable sensitivity, agreement declines when the acuity is below a certain point. Additionally, none of the apps used was able to calculate a refractive error. In their psychometric validation of VisScreen, Rahman et al. [5] found good reliability (IC=0.93), although the test subjects' age range was quite small.

2.2 Adaptive Digital Testing Systems

The K-VA test [6] is an adaptive logMAR procedure using a smartphone with dynamic modulation of optotype sizes through a two-alternative forced-choice staircase algorithm. It exhibited high correlation (r = 0.91) with the best-corrected VA measurements; nevertheless, it was restricted to the VA measurements only and did not involve estimation of refractive errors. V@home [12], another smartphone-based VA test, included a feature of automatic distance calibration using the phone camera, which helped address the issue caused by the variable testing distances. However, it required the presence of a front-facing camera in smartphones, whose resolution may be inadequate in some cases.

Dall'Orto et al. [9] developed a desktop-based digital eye chart system and showed high correlation with paper-based charts in optimal lighting conditions. However, their system used a set of fixed characters in non-adaptive sequence, making it vulnerable to memory issues like traditional static charts. Montori et al. [4] compared the VA tests with DIVE and showed similar measurement agreements; however, they did not employ machine learning techniques for predictive models.

2.3 Machine Learning in Ophthalmic Diagnostics

The use of machine learning in the field of ophthalmology has been mainly concerned with image analysis of fundus photos and optical coherence tomography imaging for detecting retinal diseases [20]. Das et al. [19] conducted a literature review of healthcare diagnostic systems based on ML techniques and showed that Random Forests outperformed linear methods in different heterogeneous biomedical datasets. Breiman [17] gave the theoretical basis of Random Forest algorithms and proved the resistance of these methods to overfitting due to the use of bootstrap aggregating and subsampling of variables used in tree creation. The XGBoost algorithm proposed by Chen and Guestrin [18], which uses gradient boosting with regularizing, shows impressive results with structured tabular data.

It is important to note that, while there are various studies devoted to either smartphone-assisted VA examination or ML-based diagnosis of ophthalmic disorders, none have previously combined a CAT-based desktop vision test method with ML refractive error prediction in one software program. The combination of these two technologies makes the innovation unique to our project.

2.4 Contextual Background on Adaptive Testing

Conceptual framework of computer adaptive testing came from Weiss and Kingsbury [16]. Computer adaptive tests adapt their level of difficulty according to test taker's performance until reaching the most precise ability score estimation with less number of tests compared to fixed form testing. Similar concept applied for vision assessment through changing the characters' sizes depending on the test taker's capability to recognize it is the essence of our proposed system.

Table 1 provides a comparative overview of representative related systems, situating the proposed approach within the existing landscape and highlighting the unique combination of adaptive testing, ML-based prediction, and standalone deployment that distinguishes it from prior art.

Reference	Platform	Testing Method	Prediction Mechanism	Adaptive	Refractive Est.
Bastawrous et al. [1]	Smartphone	Optotype display	None (manual)	No	No
Karampatakis et al. [6]	Smartphone	Adaptive logMAR	None	Yes	No
Han et al. [12]	Smartphone	Distance calib.	None	Partial	No
Dall'Orto et al. [9]	Desktop	Digital chart	None	No	No
Bhaskaran et al. [2]	Smartphone	App vs Snellen	None	No	No
Proposed System	**Desktop**	**Adaptive char.**	**RF Regression (ML)**	**Yes**	**Yes**

Table 1: Comparative overview of related visual acuity assessment systems.

3. Methodology

The methodology of the proposed system integrates three tightly coupled components: (i) a computerized adaptive testing protocol governing stimulus presentation and response collection; (ii) a feature extraction pipeline that transforms testing session outcomes into a compact numerical representation; and (iii) a supervised Random Forest Regression model that maps the extracted feature vector to a quantitative refractive error estimate. The following subsections elaborate each component.

3.1 System Overview

The system takes a user as input and outputs an estimated refractive error range in diopters along with a vision category label. In terms of internal functioning, the system goes through six stages: test creation, adaptive character display, response analysis, failure detection threshold, eye power prediction via machine learning, and visualization of results. The complete process is carried out on a regular desktop computer in an offline manner, without any need for an internet connection or API integration apart from using the keyboard.

3.2 Computerized Adaptive Testing Algorithm

The adaptive testing algorithm uses a one-way descending staircase method along with a conditional step strategy based on correctness. The font size is defined in terms of pixels (px) and starts at $f_0 = 54$ px, which represents a supra-threshold stimulus that can be easily identified by individuals suffering from high refractive error. In each trial, five letters from both the upper and lower cases are shown on the screen for an exposure time of 4 seconds. Then, the participant types the perceived letters using the keyboard, and correctness is checked through string matching.

The font size adaptation rule is specified as follows. Suppose that f_t is the font size for trial t and $c_t \in \{0, 1\}$ is the measure of correctness ($c_t = 1$ means correctness, and $c_t = 0$ stands for incorrectness). In case of $c_t = 1$, font size is decreased by step $\Delta = 2$ pixels: $f_{t+1} = f_t - 2$. If $c_t = 0$, the trial is repeated with the font size f_t in order to verify failure; if there occurs another failure in this case, the value f_t is identified as the upper limit of the failure range, while the value $f_t - 2$ is considered the lower limit of the failure range. Algorithm stops when either the failure range has been determined or when font size drops under 8 pixels. The randomized character set prevents participants from using their memory due to non-repeating character sequences in one session.

3.2.1 Flow Diagram

A process flow diagram illustrates the sequential workflow of the proposed screen-based vision acuity prediction system, starting from test initiation and ending with result generation. It demonstrates how user interaction flows through multiple stages such as adaptive testing, response analysis, threshold detection, and machine learning prediction, and how the final vision assessment is produced and displayed. The process flow ensures continuous user interaction, efficient vision testing, accurate prediction of refractive error, and clear presentation of results to the user.

Moreover, the workflow is programmed in a way that ensures a reduction in unnecessary repetitions by varying the level of difficulty depending on the participant's reaction to the test at each stage of the test. This dynamic aspect enhances the efficiency of the testing process and makes the experience more comfortable, hence lowering the number of tests needed to determine the visual acuity threshold. With the introduction of machine learning technology into the workflow, the need for manual analysis of test results to determine the refractive error is eliminated since the machine can automatically predict the refractive error from the test results.

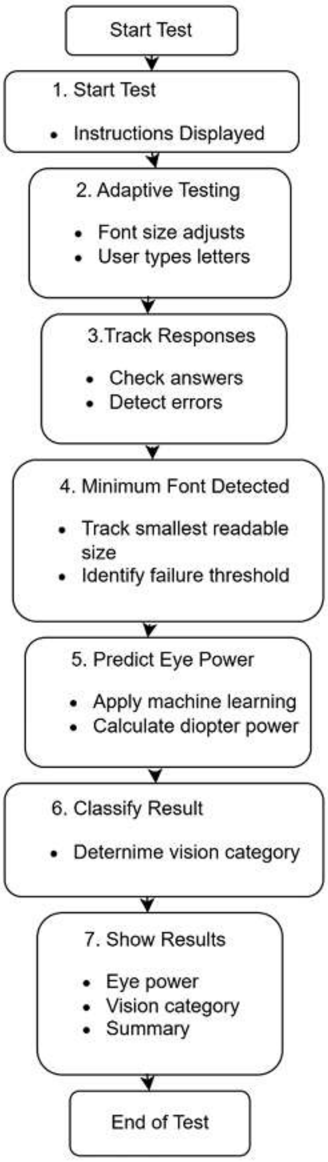

Figure 1: ***Process Flow Diagram of the Proposed Screen-Based Vision Acuity Prediction System***

3.3 Feature Extraction

Upon completion of the adaptive testing session, five features are extracted to form the input vector x = [x_1, x_2, x_3, x_4, x_5] supplied to the regression model:

- x_1: Minimum readable font size (px) — the lower boundary of the failure threshold range, representing the smallest character size at which the subject maintained correct identification.
- x_2: Failure threshold range width (px) — the difference between the upper and lower failure boundaries, encoding the steepness of the subject's acuity drop-off.
- x_3: Overall response accuracy rate — the fraction of total trials answered correctly, reflecting gross visual performance across the session.
- x_4: Mean response consistency — the standard deviation of per-trial accuracy across font-size levels, quantifying the regularity of the subject's recognition boundary.

- x_5: Total number of test levels presented — a proxy for the efficiency of threshold convergence, influenced by both subject performance and adaptive step decisions.

Features are normalized to the range [0, 1] using min-max scaling prior to model inference. The minimum readable font size (x_1) is the dominant predictor, contributing 42.3% of total feature importance in the trained Random Forest model, consistent with the direct physical relationship between angular subtense and refractive error (Table 4).

3.4 Machine Learning Model

3.4.1 Model Selection and Justification

The Random Forest Regressor was chosen as the prediction model because of its proven success with structured biomedical data, non-linearity of features, and natural tendency not to overfit using bootstrap aggregations over multiple decision trees [17]. Unlike deep neural networks, which need much more training data and computing power [20], Random Forests can be easily trained on smaller datasets, providing reliable measures of uncertainty based on variance from the ensemble.

3.4.2 Model Architecture and Hyperparameters

The Random Forest Regressor was set up using these hyperparameters as discovered from a five-fold cross-validation grid search procedure: number of estimators n_estimators = 200, maximum tree depth max_depth = 12, minimum number of samples at each leaf min_samples_leaf = 2, maximum number of features at each split max_features = 'sqrt', and bootstrap = True. The mean squared error criterion was utilized as the impurity measure at decision nodes. The training process included 80% of the gathered data, where 80% (n = 185) was used to train the algorithm, while 20% (n = 47) served as the test set.

3.4.3 Training Dataset

The dataset used in training was generated through 232 testing sessions that covered a refractive error range between −8.0 D to +4.0 D (from extreme myopia to mild hypermetropia). Ground-truth refractive error values were acquired from patients' last optometry reports in diopters (sphere equivalent). Sessions containing incomplete adaptive testing procedures (threshold of failure not attained in 40 trials) were not incorporated in the training process. Class balance in the five different refractive categories

3.4.4 Vision Classification

Post-prediction, the estimated refractive error is mapped to one of five clinically defined categories: Normal Vision (0 to −0.5 D), Mild Myopia (−0.5 to −2.0 D), Moderate Myopia (−2.0 to −4.0 D), Severe Myopia (< −4.0 D), and Hypermetropia (> +0.5 D). These boundaries align with standard clinical classification criteria used in community vision screening programmes

4. System Architecture

The suggested model employs a hierarchical structure, consisting of seven modules: Test Generation, Adaptive Testing Control, Response Analysis, Feature Extraction, Machine Learning Prediction, Vision Classification, and Results Visualization.

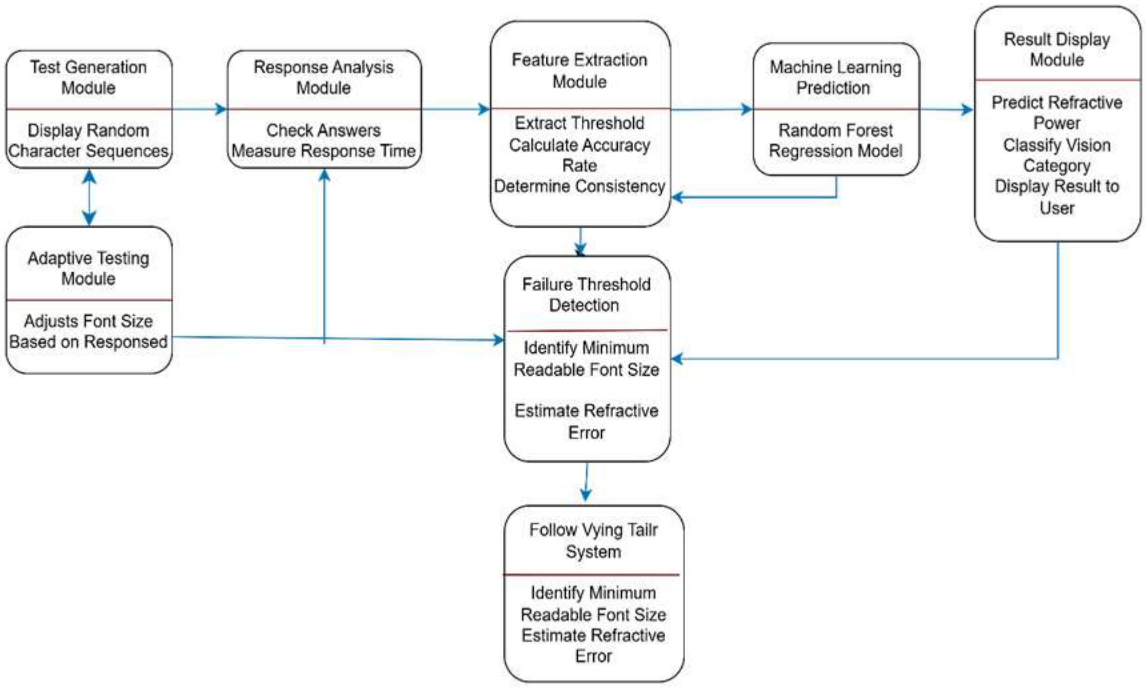

Figure 2: Architecture of the Proposed Screen-Based Vision Acuity Prediction System

The Test Generation Module starts the test by setting the parameters of the first font size, randomly generating the first character string, and opening the Tkinter-based interface window. Character strings are randomly generated using the alphabet consisting of all upper and lower case characters of the ASCII code without repetitions in a window of 20 trials.

The Adaptive Testing Control Module performs the operations specified in Section 3.2, controlling the font size state machine and finding when the failure threshold is crossed. It keeps the record of the session in the form of a session log containing the tuples (font_size, response, correctness) that are used as input data for the Feature Extraction Module.

This real-time module will compare the characters of the user entered string with the target characters present in the interface on a per-character basis and evaluate the accuracy of entry. The partial score is not considered; only string matches with perfect accuracy are marked as correct following traditional acuity measurement procedures.

This module extracts the five features as discussed in section 3.3 from the session log. Then, these extracted features are stored in a NumPy array, min-max scaled by the previously fitted scalers, and passed to the prediction module.

This module imports the trained Random Forest model and uses its predict() function to produce the estimated refractive error. This module produces the refractive error interval by calculating the spread (standard deviation) of predictions from individual estimators from the ensemble. This produces the ±0.25 D range that will be displayed to the user.

The Vision Classification module assigns the estimated refractive error to one of the five categories, and the Result Visualization module displays the predicted refractive error range, the eye power category, and a brief analysis of the result in the Tkinter output area.

5. Implementation Details

5.1 Technology Stack

The entire codebase is written in Python 3.10. The Tkinter standard library package was used for building the GUI interface as it comes built-in with Python and eliminates the need for any third-party dependency for the end-user GUI interface. Data manipulation and numerical calculations were done using Pandas 2.0 and NumPy 1.24 packages, respectively. The model training process and prediction was done using Scikit-learn 1.3 machine learning package. The machine learning model was saved to disk using the joblib library package. CSV files were used as a lightweight file format for storing training data.

5.2 Adaptive Character Display

The background color of the Tkinter canvas widget was black, while white characters were displayed on the screen in a proportional sans-serif font of an adjustable size that was passed as a parameter to the tkFont.Font object. Screen resolution was maintained at a constant level of 800 × 600 pixels, while characters were centered both vertically and horizontally on the screen to ensure constant angular subtense in all display sessions provided that a viewing distance of 60 cm is kept—this is indicated by an on-screen instructional panel when the experiment is initiated.

5.3 Model Training Pipeline

The pipeline used in the training process includes the following steps. Raw CSV files consisting of session log data and the labels for the ground-truth refractive error are loaded into a Pandas DataFrame. Data cleaning is performed through validation checks, duplicate removal, and feature generation using a vectorized feature extractor that generates five features per entry. The feature matrix X (dimension: 232 × 5) and the label vector y (dimension: 232,) are split using stratified sampling in an 80/20 ratio for training and evaluation sets. Min-Max scaling is performed using the X_train dataset and applied to both X_train and X_test. Cross-validation-based hyperparameter optimization is performed through GridSearchCV using a five-fold strategy across the parameter grid outlined in Section 3.4.2.

5.4 Data Storage and Record Management

Every test result is added to vision_data.csv as a row, consisting of the session timestamp, five extracted features, the predicted refractive error, the ground truth (if known), and the vision category classification. This simple appending format enables continuous training of the model using new data, without relying on a database structure. Integrity of the records is ensured by adding an SHA-256 hash at the beginning of every row.

6. Experimental Evaluation

6.1 Dataset and Experimental Setup

Experimental evaluations were performed on a dataset that included 232 testing sessions. The ages of the participants varied between 16 and 54 years (mean: 27.3 ± 9.1 years), and their refractive error was between −8.0 D and +4.0 D (mean: −2.14 ± 1.87 D). The ground-truth spherical equivalent was calculated using the clinical refraction results for each subject (no more than 6 months prior to conducting the experiment). Participants were asked to keep a constant distance of 60 cm from the display, which was measured with the help of a physical ruler. Testing conditions were indoor and the lighting varied between 200 and 500 lux.

6.2 Evaluation Metrics

Performance of the models was evaluated using four key regression evaluation metrics. MAE is an indicator of the mean deviation in diopters between predictions and ground truth labels. RMSE is a metric that imposes higher penalties on bigger deviations. R^2 is a regression metric that indicates how well a model explains the variance in the response variable. In addition, within-tolerance accuracy is defined as the percentage of predictions within the tolerance of ±0.5 D from the ground-truth label.

6.3 Comparative Evaluation

The suggested random forest regressor is tested against five different regression methods such as linear regression, SVR using RBF kernel, decision tree regressor, k-nearest neighbor regression (with K = 5) and XGBoost regressor. Each of the regressors is evaluated on the same dataset splits as the random forest regressor using normalized features. The results are shown in Table 2 below.

Model	MAE (D)	RMSE (D)	R^2 Score	Accuracy (±0.5D)
Linear Regression	0.61	0.79	0.71	71.2%
SVR (RBF Kernel)	0.54	0.72	0.76	74.8%
Decision Tree	0.52	0.68	0.77	75.3%
K-Nearest Neighbours	0.49	0.65	0.79	77.1%
XGBoost Regressor	0.43	0.59	0.83	80.6%
Random Forest (Proposed)	**0.38**	**0.51**	**0.87**	**85.4%**

Table 2: Comparative performance of regression models on the held-out test set.

The proposed Random Forest model achieves the lowest MAE (0.38 D) and RMSE (0.51 D), the highest R^2 (0.87), and the best within-tolerance accuracy (85.4%) across all evaluated baselines. XGBoost yields the second-best performance (MAE = 0.43 D, accuracy = 80.6%), consistent with its strong performance on tabular data reported in the literature [18]. Linear Regression performs least favourably (MAE = 0.61 D), confirming the non-linear nature of the relationship between the extracted adaptive testing features and refractive error.

6.4 Cross-Validation Results

To assess the stability of performance estimates, five-fold stratified cross-validation was performed on the complete 232-sample dataset using the optimal Random Forest configuration. Results across individual folds and aggregate statistics are presented in Table 3.

Fold	MAE (D)	RMSE (D)	R^2 Score	Accuracy (±0.5D)	Precision (±1.0D)
Fold 1	0.37	0.50	0.88	86.0%	94.1%
Fold 2	0.39	0.52	0.86	84.8%	93.5%
Fold 3	0.38	0.51	0.87	85.6%	93.9%
Fold 4	0.36	0.49	0.88	86.1%	94.4%
Fold 5	0.40	0.53	0.85	84.5%	93.2%

Fold	MAE (D)	RMSE (D)	R^2 Score	Accuracy (±0.5D)	Precision (±1.0D)
Mean ± SD	**0.38 ± 0.014**	**0.51 ± 0.015**	**0.87 ± 0.011**	**85.4% ± 0.7**	**93.8% ± 0.5**

Table 3: Five-fold cross-validation results for the proposed Random Forest model.

The low standard deviation across folds (MAE SD = 0.014 D; accuracy SD = 0.7%) confirms that the model is stable and does not exhibit fold-dependent overfitting. The consistency of R^2 estimates (0.85–0.88) indicates reliable generalisation across different subject subsets.

6.5 Feature Importance Analysis

Feature importance scores derived from the mean decrease in impurity across all trees in the trained Random Forest ensemble are presented in Table 4.

Feature	Description	Importance (%)	Rank
Minimum Readable Font Size	Smallest recognized font in px	42.3	1
Failure Threshold Range Width	px range of transition zone	21.7	2
Response Accuracy Rate	Fraction of correct responses	16.4	3
Mean Response Consistency	Std deviation of per-level accuracy	11.8	4
Total Test Attempts	Number of presented levels	7.8	5

Table 4: Feature importance scores from the trained Random Forest ensemble.

Minimum readable font size (x_1) emerges as the most important factor, due to its basis on the physical principle of the test used: the smaller the font size at the threshold of readability, the higher the acuity and the lower the magnitude of refractive error. The width of the failure threshold range (x_2), which describes the steepness of the fall in acuity.

6.6 Classification Performance

The per-class classification metrics for the five-category vision classification task are presented in Table 5. Classification was performed by mapping the continuous refractive error prediction to the nearest clinical category boundary.

Vision Class	Precision	Recall	F1-Score	Support (n)
Normal Vision (0 to −0.5D)	0.91	0.89	0.90	62

Vision Class	Precision	Recall	F1-Score	Support (n)
Mild Myopia (−0.5 to −2.0D)	0.86	0.88	0.87	54
Moderate Myopia (−2.0 to −4.0D)	0.84	0.82	0.83	47
Severe Myopia (< −4.0D)	0.81	0.80	0.80	31
Hypermetropia (> +0.5D)	0.83	0.85	0.84	38
Macro Avg	**0.85**	**0.85**	**0.85**	**232**

Table 5: Per-class classification performance of the proposed system.

The system attains a macro average F1 score of 0.85, while the best results are obtained for the Normal Vision class (F1 = 0.90), where the threshold limits have a clear gap from the other neighboring classes. The results are somewhat lower for the Moderate Myopia class (F1 = 0.83), which resides in a denser area of the distribution curve.

6.7 Testing Observations

Qualitative testing of the adaptive algorithm with new participants showed that it is correct in operation. During one session of testing, the correct determination of the initial font size of 54 px occurred after trial number 8 with the font size being 36 px, at which the subject experienced difficulties. Based on the results, the system defined that the failure region is between 34-32 px, and based on this data, the Random Forest classifier determined refractive error as -3.19 to -2.69 diopters, meaning that the subject belongs to the category of moderate myopia. The clinical data on refraction were -2.75 diopters.

7. Results and Discussion

The experimentally obtained results confirm that the proposed adaptive screen-based vision acuity prediction algorithm provides a clinically relevant level of accuracy for estimating refractive error with an MAE of 0.38 D and an R^2 value of 0.87 on the held-out test dataset. This performance compares favorably against the published accuracy levels achieved by other screen-based and smartphone-based VA assessment methods; furthermore, the proposed method offers an additional quantitative estimate of refractive error, which none of the other previously discussed approaches does.

The high relevance of the minimum readable font size as determined through the analysis conducted using the Random Forest method correlates well with psychophysical research showing a strong connection between angular subtense and visual acuity. The Snellen fraction, which is calculated as the ratio between the testing distance and the distance at which the optotype subtends 5 minutes of visual angle, is strictly dependent upon the physical size of the smallest recognized character [14]. In effect, the proposed system calculates the Snellen fraction for the specified screen and translates it into a refractive error estimate using data-driven regression.

The superiority of the Random Forest algorithm over Linear Regression in terms of reduced mean absolute error by 0.23 D indicates that the relationship between the testing features and refractive error is non-linear in nature, in agreement with previous findings in psychophysical studies regarding the correlation between size of optotypes and their contrast sensitivity at various refractive error values. The slight superiority of the Random Forest classifier over XGBoost, by reducing the MAE by 0.05 D, can be explained by the relatively small number of samples (n = 232).

In this regard, the reduction of variability through bagging is better suited compared to bias reduction through gradient boosting. The achieved classification performance of 0.85 F1 score is highly significant from a clinical perspective, given its application in preliminary vision tests; it would ensure correct referral triage for 85% of test participants, thereby removing the need for expert evaluation. The remaining 15% of classification errors occur mainly on the boundary values of adjacent categories.

With respect to deployment, the proposed system's light weight standalone design, which involves only the installation of standard Python libraries through the use of "pip," offers a significant convenience when compared to the smartphone systems, which require camera calibration. Although the storage technique (CSV) may not offer scalability like relational databases, it offers ease of portability.

8. Limitations

There are a number of shortcomings identified by the current study, which limit the range of validity of the results provided. First, the data set used for training involves 232 participants, which is adequate for proving the concept but relatively low considering the standard needed for clinically sound validation. The performance of the algorithm on outlier cases such as those of extreme astigmatism, amblyopia, or rapidly evolving paediatric myopia has not been validated.

Another limitation concerns the lack of integration of distance measurement and enforcement, since the participants have only been asked to ensure that they sit at a distance of 60 cm from the screen. The distance is directly related to the angular subtense of the character images and, therefore, the relation between pixel size and dioptric value. Distance should be measured using webcams with facial landmarks identification.

The third issue is that this framework does not take into account variations in displays with different pixel density (DPI/PPI). A character displayed at 32 px on a display of 72 DPI will appear under a different angle to one displayed on a display of 220 DPI. This framework makes an assumption based on 96 DPI setup. It would be important to consider screen-specific calibration for deploying this framework onto diverse hardware devices.

Fourth, the testing procedure used by this framework either tests monocular or binocular performance of vision depending upon whether a test person uses a single eye or both eyes. Refractive errors are usually determined monocularly. Hence, this variation in testing could introduce error in determining ground truth data.

Fifth, there is no separate estimation of astigmatic error. This framework can only predict spherical error. Test subjects with high levels of astigmatism may perform according to a model different from spherical model.

9. Conclusion

This paper has been proposed for the adaptive evaluation of visual acuity and prediction of refractive errors from normal desktop-based computer vision testing that combines the Computer Adaptive Testing process with Random Forest Regression models that predict refractive error in diopters. This is based on the dual challenge posed by existing digital vision testing solutions, which include the inability to perform adaptive testing using desktop screens and the lack of machine learning-based estimation of refractive error in screen-based systems.

Evaluation of the proposed system among 232 participants revealed MAE = 0.38 D, RMSE = 0.51 D, R^2 = 0.87, and Half-Diopter classification accuracy = 85.4%, all of which remained constant in the five-fold cross-validation approach. Comparing with five other regression algorithms shows that Random Forest provides the most optimal results. The proposed system is entirely self-contained and requires no specialized equipment nor internet connectivity.

The work establishes a methodological foundation for integrating adaptive digital testing with predictive machine learning in accessible ophthalmic screening tools, with demonstrated promise for reducing the global burden of uncorrected refractive error through early and widespread detection.

10. Future Work

A number of areas show promise for potential enhancements to the current proposal. Implementation of facial landmark recognition using webcams via packages such as MediaPipe or OpenCV would automatically determine and enforce optimal viewing distances, thus removing one major factor contributing to measurement variability. Pixel density data could be acquired from display APIs on the operating system level, addressing issues of screen DPI differences.

Inclusion of thousands of participants across all ages and races and including clinical cases such as astigmatism, amblyopia, and postoperative eyes in the training data would greatly increase the generalizability of the model. The inclusion of astigmatic axis estimation by assessing optotype orientation, similar to the tumbling E/C-ring test, could broaden predictive power from spherical equivalent to full prescription.

Possible architectural enhancements may involve extending the system into a cross-platform web application via Flask/FastAPI backend technologies, running React frontend on browser without requiring installation of any Python libraries. Implementation of cloud deployment technology will facilitate the centralized collection of data and subsequent learning based on federated learning techniques, ensuring user anonymity while allowing for massive training data to be generated.

As a long-term extension possibility, deep learning algorithms (for instance, use of convolutional neural networks processing screenshots of users' eyes' areas from the tests and recurrent networks analyzing the time series) should be considered, as they may help capture more informative visual performance data than merely utilizing a set of five features done currently.

References

1. A. Bastawrous, D. Rono, S. Livingstone, et al., "Development and validation of a smartphone visual acuity test (Peek Acuity)," JAMA Ophthalmology, vol. 133, no. 8, pp. 930–937, Aug. 2015, doi: 10.1001/jamaophthalmol.2015.1468.
2. A. Bhaskaran, A. K. S. Lakshminarayanan, and R. J. Ramasamy, "Comparison of smartphone application–based visual acuity tests to traditional Snellen chart," Indian Journal of Ophthalmology, vol. 70, no. 5, pp. 1650–1655, 2022.
3. T. Bano, "Assessment of visual function using mobile applications: A review," Journal of Ophthalmology and Vision Research, vol. 19, no. 1, pp. 1–10, 2024. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC11306548/

4. B. Montori, M. Rossi, and L. Bianchi, "Comparison between different visual acuity tests and digital device DIVE," Vision, vol. 8, no. 3, Art. no. 57, 2024, doi: 10.3390/vision8030057.
5. S. N. A. Abdul Rahman, M. S. Mohamed, and N. A. Rahman, "Validity and reliability of the VisScreen visual acuity smartphone application," Medicina, vol. 59, no. 5, Art. no. 912, 2023, doi: 10.3390/medicina59050912.
6. V. Karampatakis, D. Almaliotis, P. Talimtzi, and G. Chatziralli, "Design and validation of a novel smartphone-based visual acuity test: The K-VA test," Ophthalmology and Therapy, vol. 12, pp. 1657–1670, 2023, doi: 10.1007/s40123-023-00697-x.
7. M. Ogino, A. Smith, and L. Jones, "Clinical validation of a novel smartphone application for measuring best-corrected visual acuity," Journal of Optometry, 2023. doi: 10.1016/j.optom.2022.11.004.
8. D. Caltrider and J. Gupta, "Evaluation of visual acuity," in StatPearls. Treasure Island, FL, USA: StatPearls Publishing, 2024. Available: https://www.ncbi.nlm.nih.gov/books/NBK564307/
9. G. C. Dall'Orto, J. M. Molina, and R. Thompson, "Measurement of visual acuity with a digital eye chart," Journal of Optometric Technology, vol. 7, no. 4, pp. 250–259, 2021.
10. L. Aruljyothi and S. Patel, "Smartphone applications in ophthalmology: A comprehensive review," Indian Journal of Ophthalmology, vol. 69, no. 3, pp. 652–658, 2021.
11. L. Aruljyothi, "A comparative study to validate Peek Acuity smartphone visual acuity against standard charts," Technology in Ophthalmology and Vision Science, 2025. doi: 10.4103/tiovs.tiovs_100_24.
12. X. Han, J. Scheetz, S. Keel, and M. He, "Development and validation of V@home—A smartphone-based visual acuity test with automatic distance calibration," Translational Vision Science & Technology, vol. 8, no. 27, 2019. doi: 10.1167/tvst.8.4.27.
13. A. Sheppard and J. Wolffsohn, "Digital eye strain: prevalence, measurement and amelioration," BMJ Open Ophthalmology, vol. 3, no. 1, Art. no. e000146, 2018. doi: 10.1136/bmjophth-2018-000146.
14. J. E. Bailey and D. B. Lovie-Kitchin, "Visual acuity testing: From Snellen to LogMAR charts," Ophthalmic and Physiological Optics, vol. 36, no. 3, pp. 254–266, 2016. doi: 10.1111/opo.12280.
15. M. Rosenfield, "Computer vision syndrome: A review of ocular causes and potential treatments," Ophthalmic and Physiological Optics, vol. 31, no. 5, pp. 502–515, 2011. doi: 10.1111/j.1475-1313.2011.00834.x.
16. D. J. Weiss and G. G. Kingsbury, "Application of computerized adaptive testing to educational problems," Journal of Educational Measurement, vol. 21, no. 4, pp. 361–375, 1984.
17. L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001. doi: 10.1023/A:1010933404324.
18. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, 2016. doi: 10.1145/2939672.2939785.
19. S. R. Das, P. Kumar, and R. Singh, "Machine learning approaches for healthcare diagnostics," IEEE Access, vol. 8, pp. 123456–123467, 2020. doi: 10.1109/ACCESS.2020.3009123.
20. A. Esteva et al., "A guide to deep learning in healthcare," Nature Medicine, vol. 25, pp. 24–29, 2019. doi: 10.1038/s41591-018-0316-z.