American University of Sharjah

Fall 2024

College of Engineering

Department of Computer Science and Engineering

COE 476 - Neural Networks and Deep Learning

Classification of Diamond's Cut, Carat, Color, Clarity, and Price Prediction

Mohammed Ali - b0088658

Ezekiel Morata - b00089094

Mihir Dinesh - b00090511

Date of Submission: 05/12/2024

Submitted to:

Dr. Imran Zualkernan

Salsabeel Shapsough

Table of Contents

0. Table of Contribution

Team member AUS ID	Team member name	Team member's contribution to the project	Percentage Effort (out of a total of 100%)
88658	Mohammed Ali	CNN model, Data Selection, Data Cleaning and Feature Engineering, Results	40%
89094	Ezekiel Morata	Validation Methodology, Tables, References, BibText File, Formatting, Helped in small parts	27%
90511	Mihir Dinesh	Description, Previous Approaches, Future Work and Conclusion, Network Architecture Selection, Why it works, Price Prediction model, Data Selection, Overall ProofReading	33%

1. Description of the Problem

Traditionally, diamond grading is a time-consuming, subjective, and often inconsistent process. It is performed manually by expert gemologists. There has been very little application of neural networks to diamond grading from images, with most existing machine learning searches focusing on price prediction.

Our team aims to create a novel deep neural network that focuses on recognizing the four Cs of round diamonds; Cut, Color, Carat, and Clarity, based on their images and predicting their prices through a multi-output 2d convolutional neural network. Our project has little to no precedent in existing research with only a few commercial companies marketing their products for diamond shopping. Hence our project can significantly lower operational costs by minimizing the need for expert graders, increasing efficiency in the grading process. Such a system would be particularly beneficial for large-scale diamond processing facilities, enabling real-time grading and sorting. A tertiary goal of our project is to make it more accessible by integrating the diamond price prediction with the grading process. Making the system more accessible allows smaller jewelry businesses, independent gemologists and diamond customers to benefit from consistent grading without the help of specialized equipment. For the team to achieve this goal, we also plan to have a diamond price prediction model that can utilize the output predictions from our dataset.

The challenges of creating a neural network capable of diamond grading stem from the lack of publicly available datasets and limited prior research on using neural networks for this specific task. The uniqueness of each diamond, combined with subtle visual differences that distinguish grades, makes it difficult to create a generalized, accurate model. Variations in lighting, angles, and background can also significantly impact the perceived attributes of a diamond. Finally, predicting the 4 C's simultaneously adds complexity, as each attribute has unique grading criteria that we want a single model to learn.

2. Previous Approaches

Work	Summary Strength/Weakness	Architecture	Dataset
[1] A 3D convolutional neural network model with multiple outputs for simultaneously estimating the reactive transport parameters of sandstone from its CT images (2024)	This model was designed to predict porosity, tortuosity, specific surface area (SSA), and permeability of sandstone. It begins with a convolutional layer, followed by a batch normalization layer, an activation layer, and a max-pooling layer. This is succeeded by multiple residual blocks—BasicBlock for ResNet-10, ResNet-18, and ResNet-34, and BottleNeck for deeper variants like ResNet-50, ResNet-101, and ResNet-152. These blocks are followed by an average pooling layer, a flattened layer, a dense layer, and a multi-output layer with a linear activation function for end-to-end prediction of the four parameters. The model is trained with 3D subsample CT images using the Adam optimizer and MAE as the loss function. ResNet-34 had the best results yielding the best MAE and R^2 for the metrics Porosity (MAE= 0.24, R^2= 0.995), Tortuosity (MAE= 0.27, R^2= 0.941), SSA (MAE= 0.37, R = 0.994) and Permeability (MAE= 30.80, R^2= 0.977) Strengths: The model effectively demonstrates multi-output regression using CNN with high prediction accuracies for every metric. It maintains its effectiveness on unseen sandstone CT images showing strong generalization ability. Weakness: The performance is highly dependent on the quality of 3D CT images, which have high computational requirements.	3D CNN (ResNet)	The dataset comprises CT 3D binary images of 11 sandstone varieties, sourced from the Digital Rocks Portal. Using a sliding window of 1283 voxels with a step of 96 pixels, a total of 11,000 cube subsample images were generated
[2] Jewelry rock discrimination as interpretable data using laser-induced breakdown spectroscopy and a convolutional LSTM deep learning algorithm (2024)	This study uses a CNN-LSTM combination model to classify various jewelry rocks, such as agate, turquoise, calcites, and azure, from different historical periods and styles associated with Shahr-e Sokhteh. It uses as input the spectra data from agate, calcite, and other semi-precious stones from the Libs experiment. The approach combines CNN layers with kernel size 3 for feature extraction with LSTMs for sequence forecasting. The Adam optimization function with a learning rate of 0.0005 and the categorical cross-entropy cost function was	CNN-LSTM architecture	A self-created dataset that includes spectra from 43 agate, 20 calcite, 59 turquoise, and 46 lapis lazuli samples, resulting in 7678

	used, with the training running for 100 epochs, a batch size of 128, and a validation split of 0.25. The CNN layers used the ReLU activation function, while the LSTM layers used the tanh activation function. The classification layers include three perceptron layers, assigning probabilities to each gemstone based on their features. The study was able to offer an accuracy of above 98%, hence being very accurate. Strengths: It effectively combined CNN for extracting features with LSTM to support sequence forecasting. Furthermore, it was able to optimize itself using Adam optimization and cost functions, thus resulting in reduced overfitting and faster training times. Weakness: The results are highly dependent on the specific dataset of jewelry rocks, which in turn need to have their spectra data analyzed by Libs. As a result, the model is highly specialized and not very usable for other activities, trading it for its high accuracy.		interpretable data points after combining and preprocessing.
[3]Ordinal Regression with Multiple Output CNN for Age Estimation (2016)	The paper proposes an End-to-End deep learning approach for age estimation using a Multiple Output Convolutional Neural Network (CNN). The network architecture consists of three convolutional layers, three local response normalization layers, two max-pooling layers, and a fully connected layer with 80 neurons. The model takes 60x60x3 color face images as input, with the final layer branching into K-1 output layers, each corresponding to a binary classification task predicting whether a sample's age is above a specific rank. This transforms an ordinal regression problem into K-1 binary classification sub-problems, where each task predicts whether a sample's age is above a specific rank. The CNN learns features and performs regression at the same time, with output layers sharing intermediate representations. They obtained MAE=3.27 ± 0.02 and MAE=3.34 ± 0.08 on the MORPH-II and AFAD datasets respectively, which were higher than the competing models at the time. Strengths: They obtained a high degree of accuracy, which was validated by splitting 100 times and	Ordinal Regression - CNN (multi-output)	They used the MORPH They made the Asian Face Age Dataset (AFAD), containing over 160,000 facial images with precise age labels by collecting facial images from the RenRen Social Network.

	averaging. Their approach also showed potential for applying a similar approach to other ordinal regression problems beyond age estimation. Weakness: They used a relatively shallow CNN with only 3 conv layers, meaning accuracy could be further increased. Additionally, the dataset was trained on Asian faces meaning that it might not be as applicable to other ethnic groups.		
[4] Gemstone Classification Using Deep Convolutional Neural Network (2024)	The paper uses a feedforward CNN, DenseNet-169 pretrained transfer learning model to classify 87 types of gemstones such as diamonds, rubies, sapphires, amethysts, etc. The topmost layer was removed with a Global Average Pooling Layer Appended (GAP) to generate feature maps for the 87 categories. The initial layers of DenseNet169 extract low-level features like edges and textures. Deeper layers extract higher-level features relevant to gemstone classification. Relu is used throughout the model with Softmax for the final layer to get the class probabilities. Strengths: The model achieved a significant 79% accuracy, surpassing existing methods. Transfer learning also ensured that the training was far more efficient than learning from scratch, with max performance reaching 20 epochs. Weaknesses: The model struggles to predict Ametrines, Green Aventurine, Emeralds, and Goshenites. The dataset also suffers from having not enough samples as each class only has around 33 images that can be used for training and even augmentation cannot effectively address the keen lack of samples.	CNN (DenseNet 169) pre-trained transfer learning model	Gemstone Images from Kaggle- 3200 images and 87 classes. With 2800 images for training and 400 for testing. Data augmentation was implemented through the Keras Image DataGenerator.
[5] Diamond Color Grading Based on Machine Vision (2009)	The paper presents a machine vision-based approach to measure 3D structures and grade diamond colors. The method integrates stereo vision for 3D measurement and a backpropagation neural network with 24 input parameters and 12 nodes for color grading in modified HSV space. Linear and non-linear features are extracted using a virtual motion control system. The model used in the paper was able to achieve a high accuracy of 90.625%. Strengths: The paper showcases the successful use of neural networks for color grading diamonds, a key component of our project. They were successful	Color grading employs a BP neural network with inputs derived from color features (Hue, Saturation) and compressed joint distributions	Diamond images were acquired using a custom-built stereo vision system integrating sphere lighting and CCD cameras. Preprocessing includes color

	and able to obtain a high accuracy in diamond color grading. Additionally, their models were also successful when tested on real diamonds. Weaknesses: The model uses a custom image acquisition system instead of images and needs complex preprocessing for noisy backgrounds. More importantly, the paper is old and with advancements in computational neural network research, newer techniques such as CNN and more layers could achieve a higher degree of accuracy.	in HSV space.	normalization and region segmentation. CAD models guide initial object positioning.
[6] GemInsight: Unleashing Random Forest for Diamond Quality Forecasting (2023)	GemInsight is a machine learning-based system for automated diamond quality forecasting using a Random Forest (RF) model. It evaluates diamonds based on the 4Cs: Carat, Cut, Color, and Clarity along with Depth, Table, Price, X, Y, and Z (dimensions) with output as 5 different quality grades. They used the diamond dataset from Kaggle on their project, which we also plan to utilize for our price prediction. The Random Forest model achieved an accuracy of 79%, outperforming the Decision Tree model's 70.8% accuracy. Strengths: The model is unique in the fact that it is one of the few approaches that try to assess the overall quality of diamonds instead of just one of cut/carat/color/clarity. However, their approach has several drawbacks. The diamond quality is determined using a combined metric instead of using the 4Cs individually which the Gemological Institute of America uses for their grading system. Their model relies on numerical and categorical data; it does not incorporate image data for visual inspection, which is crucial for assessing cut and clarity.	Random Forest ensemble model. Decision Tree was also tested but performed worse due to overfitting. The system uses classification for quality accession.	Dataset sourced from Kaggle, containing 53,940 records. Features include Carat, Cut, Color, Clarity, Depth, Price and dimension
[7] Enhancing the performance of transferred EfficientNet models in leaf image-based plant disease classification	The paper explores the enhancement of EfficientNet architectures for automated plant disease classification using leaf images. The researchers implemented a transfer learning approach, utilizing pre-trained Noisy-Student weights from ImageNet to fine-tune the EfficientNet models on the PlantVillage dataset, which was expanded and augmented to improve feature diversity. The training process used learning rate schedulers and early stopping, to mitigate overfitting and optimize	Efficient Net with transfer learning using pre-trained Noisy-Student weights from ImageNet	The PlantVillage dataset, consisting of 54,305 original leaf images across 14 plant species, was augmented to

	training efficiency. The study compared multiple EfficientNet variants (B0 to B7) and evaluated their performance across both augmented and non-augmented datasets. EfficientNet-B3 achieved the highest accuracy (99.997%) on the non-augmented dataset, while EfficientNet-B5 excelled on the augmented dataset with the same accuracy, surpassing other state-of-the-art models like AlexNet, VGG, and ResNet as well as other CNNs. The results show the potential of EfficientNet for image classification. Strengths: The study showcases exceptional classification accuracy (99.997%) through advanced transfer learning and model fine-tuning techniques. They also conducted multiple validation tests for overfitting. Weakness: The model relies heavily on the PlantVillage dataset, which may not represent real-world variations (e.g., lighting, background clutter) accurately. This limits generalizability to real-time, field-based applications.		61,486 images to increase the dataset's robustness and feature variety.
[8] GemID: A Hybrid CNN-Random Forest Approach for Accurate Gemstone Identification	This paper proposes a model to identify 6 different gemstones: sapphire, ruby, emerald, amethyst, topaz, and diamond by combining CNN with RF for classification. The gemstone feature extraction model starts with two convolutional layers to detect patterns, followed by a max-pooling layer to reduce data complexity. After another 2 convolutional and 1 max-pooling layer to refine feature extraction, a flattening process converts the data into a 1D vector. Finally, a Random Forest classifier accurately categorizes the gemstones based on these extracted features. They intended to balance depth and computational efficiency. Strengths: They were able to successfully combine 2 different models and obtain a model with good accuracy. The input is also only images instead of having additional complex signals like CAD models or spectra, ensuring the model has a wider range of applicability. Their dataset also has a robust number of samples when compared to other models. Weaknesses: Their model is not as accurate as other sources such as the model using DenseNet-169. This could be due to the relative lack of layers in their	CNN-Random Forest hybrid model	A custom dataset was assembled from secondary sources using gemstone photos which were further augmented.

	CNN. Adding more layers so that it can extract additional features and refine it would improve the quality of the model.		
[9] Inclusion extraction from diamond clarity images based on the analysis of diamond optical properties (2019)	This paper focuses on a more manual approach, developing an inclusion extraction algorithm for diamond clarity from images. It uses images of diamonds captured through a custom image acquisition system. It then employs image processing techniques to accurately separate regions of interest and identify inclusion regions. The algorithm got a 92% match rate when compared to manual evaluations by gemologists. Strengths: It has high accuracy, robustness to noise, and detailed consideration of diamond optical properties. Weakness: Their reliance on a manual approach means that they have to rely on manual feature engineering. Hence using CNNs could potentially enhance the accuracy and efficiency of inclusion extraction since they automatically learn and adapt to extract relevant features from the image. Additionally using CNNs could also fix the problems of the algorithm where it struggles with overlapping regions of high intensity and complex reflections in poorly cut diamonds. Lastly, their algorithm requires the use of the image acquisition system for inputs.	None applicable (Algorithm was developed manually based on principles of physics)	The dataset for testing consists of 150 diamond samples with various clarity grades.
[10] An intelligent system for mineral identification in thin sections based on a cascade approach (2016)	This article proposes an intelligent system for mineral identification in thin sections using a cascade approach. The system has 2 phases: segmentation using color parameters and an incremental clustering algorithm to produce mineral clusters, and identification using a cascade classification approach with color-based identification at Level 1 and texture-based identification at Level 2. 23 different ANNs corresponding to the number of input minerals are trained separately by using the color components In the second level of the cascade, another ANN is trained for all minerals labeled as the group#2 based on the texture features. With this approach, they were able to get an overall accuracy of 93.81%. Strengths: They were able to obtain a high accuracy for their model, beyond other techniques available at	ANN-based- 23 different ANNs for color components corresponding to 23 minerals in the dataset and a single ANN for texture features.	The dataset consists of 135 thin-section images of glass and 22 common igneous minerals, captured in both planes and cross-polarize d light.

	the time. Additionally, they showcased a unique way of being able to integrate both texture and color to distinguish minerals Weaknesses: For image recognition they used 23 different ANNs for their model. ANNs are inferior to CNNs for image processing and feature detection. Additionally, it is computationally expensive to train and use separate models for each mineral instead of 1 model for multiple. Hence using a CNN here to train for colour recognition would be vastly more useful.		
[11] Mineral Classification Using Machine Learning and Images of Microscopic Rock Thin Section (2019)	This study focuses on automating mineral type classification from rock-thin sections using digital image processing and simpler machine learning techniques like nearest neighbor and decision trees. The method involves capturing mineral images under cross and plane-polarized light, extracting optical properties such as color and texture, and applying machine learning algorithms. Results showed high accuracy, achieving up to 97% and 93% for the respective datasets, showing that simpler machine learning algorithms can be highly effective, potentially rivaling more complex neural networks. Strengths: They were able to achieve a high accuracy while using simpler approaches allowing for easy implementation. Weaknesses: They need to manually extract features. The dataset does not have many samples indicative of real-life scenarios as it consists of specialized images captured under cross and plane-polarized light through a microscope.	The study employed K-nearest neighbors and decision tree algorithms for classification.	Two datasets were used: one from Ferdowsi University of Mashhad with 17 minerals, and another built by hand with 4 minerals. They consist of images captured under cross and plane-polarized light through specialized microscopes.
[12] Gemstone classification using ConvNet with transfer learning and fine-tuning (2022)	This study uses transfer learning (TL) to enhance the performance and reduce the training time of a DL for classifying gemstones, using InceptionV3 as the base model. The model enhancement involved adding an Average Pooling Layer to the output layer to sample each square of the feature map to its average. Additionally, a Dense Layer with Softmax activation was incorporated to normalize the output, providing a probability distribution over predicted classes. Three Keras callback methods were implemented: reducing the learning rate on a plateau, early stopping, and Model Checkpoint.	CNN with transfer learning using InceptionV3	Gemstone Images from Kaggle- 3200 images and 87 classes. With 2800 images for training and 400 for testing.

	RMSProp optimization was used with a learning rate of 10^{-4}, 100 epochs, and a batch size of 64. The best configuration involved freezing the first 200 layers of InceptionV3 and fine-tuning the final 111 layers. The model was able to achieve an accuracy of 72%. Strengths: The TL approach using InceptionV3 model, was able to achieve high accuracy while also reducing training time. Weakness: The model struggles with high similarity between certain gemstone classes like Beryl Golden, Sapphire Yellow, and Chrysoberyl leading to misclassifications. This could be a limitation of using transfer learning. Training a model from scratch for this task could further improve performance.		
[13] Jewelry Recognition via Encoder-Decoder Models (2023)	The paper explores jewelry recognition through image captioning using encoder-decoder architectures. It consists of using different image captioning models to detect the jewels from an image and generate a natural language description of the accessory. Then, this description is also utilized to classify the accessories at different levels of detail. The generated caption includes details such as the type of jewel, color, material, and design, which is similar to what we need to gather from our dataset. The encoder consists of CNNs, while the decoder employs RNNs. Transfer learning is leveraged using pre-trained CNNs (VGG-16, InceptionV3, and MobileNet), and the decoders are either LSTMs or Gated Recurrent Units (GRU). The dataset, comprising 2,687 images, was augmented using techniques like rotations, shifts, and color adjustments. Models were trained, validated, and tested with respective splits of 75%, 15%, and 10%. The best configuration, VGG-16 with GRU, using 256 neurons, Adam optimizer, batch size of 16, and a learning rate of 0.001, achieved a captioning accuracy of 95%. Strengths: The best model achieved 95% captioning accuracy, demonstrating robust performance in detailed image descriptions. Weakness: The relatively small dataset of 2,687 images for an NLP task will limit the model's ability	The study uses multiple models along with transfer learning. Encoders use CNNs (VGG-16, InceptionV3, MobileNet). Decoders use RNNs (LSTM and GRU).	Images were sourced from two online jewelry stores in Córdoba, Spain. After augmentation, the dataset contained 2,687 images divided into training (75%), validation (15%), and testing (10%) sets.

	to generalize across diverse jewelry types.		
[14] A Comparison of the Efficiency of Using a Deep CNN Approach with Other Common Regression Methods for the Prediction of EGFR Expression in Glioblastoma Patients (2019)	This study compares traditional and deep learning approaches for predicting molecular features from glioma MRI images. A CNN was used to segment the glioma substructures (whole tumor, tumor core, enhancing core) and extract high-dimensional features, which were reduced through a three-step process involving median absolute deviation (MAD), correlation filtering, and Boruta feature selection, ultimately retaining 358 features. Elastic Net regression, combining Lasso and Ridge regularization, was employed to address high-dimensionality issues, optimizing λ values via cross-validation. The CNN-based regression model, bypassing feature selection, outperformed traditional models by learning directly from the raw features. Comparing both approaches revealed that the CNN without feature selection had better efficiency, with training and test loss values of 5.94 and 2.903. Strength: The deep CNN approach significantly reduced the loss value (5.94) compared to Elastic Net (10.97) and Lasso (15.53), demonstrating superior accuracy in EGFR expression prediction from high-dimensional MRI data. It utilized multiple MRI modalities and anatomical planes, enhancing the model's robustness and reducing dependency on pre-defined handcrafted features. Weakness: The relatively small dataset (N = 166) may restrict the generalizability of the model, although it demonstrated good results despite this limitation. Due to this, it risks overfitting because of the vast number of extracted features (4,915,200 per patient), especially with a limited dataset.	CNN with 4 layers including input	The imaging dataset included 5235 MRI scans for 262 patients diagnosed with GBM and were downloaded from The Cancer Imaging Archive. The EGFR expression data, for the corresponding patients in the TCIA dataset, were downloaded from The Cancer Genome Atlas (TCGA). 166 patients had both the necessary imaging and genomic data
[15] An Exploration of Deep Learning-Based Object Detection and Classification in Yakshagana Imagery with YOLOv3 (2024)	YOLOv3 neural network architecture is a real-time object detection system that can recognize specific things in videos, live footage, and images for object detection and classification in Yakshagana imagery; "persons," "jewelry (Yedekattu)," and "crowns (kireeta)" to classify within the dataset and fitting the model with batch size of 64, a threshold of 0.5, using Adam and SGD optimizers with a learning rate of 0.001, 100 epochs, and a dropout layer of 0.5 rate. A reason for the neural network to be	You Only Look Once version 3 (YOLOv3) algorithm for object recognition.	The dataset of their research consists of 10,000 images of Yakshagana characters manually created for the model.

	considered for our project was the ability of its single-pass detection, as unlike regular CNNs, YOLOv3 processes the images with a single pass for detection, reducing computational overhead. The model also classifies images based on the shapes of crowns and jewelry. Additionally, it does not use specialized equipment or spectra and relies on images similar to the images in our model. Strengths: The model achieved a mean average precision of 93.83% showcasing strong performance in objection detection. Intricate and distinctive elements in Yakshagana imagery, with various shapes of crowns and jewelry styles, were successfully recognized by the model. Weakness: Unfortunately, there is a tradeoff between speed and accuracy, as it did not achieve high recall values. The YOLOv3 architecture also struggles to recognize small-scale objects.		
[16] Amber Gemstones Colour Classification by CNN (2023)	This study shows how a CNN can categorize amber gemstones based on their multiple colors with two different strategies in mind. The study of the paper can relate to our project both intend to recognize the color of the gemstones: Baltic amber gemstones for them and diamond gemstones for us. The network used the 5-fold approach to minimize the categorical cross-entropy loss function and apply the dice score component to achieve a 90% accuracy and adjust the training loss function. The second approach was splitting the images into 60x60 px patches and using the majority voting rule to predict the classes, resulting in 89.6% accuracy. The base CNN structure comprises 3 Conv2D layers with a max-pooling layer after each one, a dense layer, and an output layer with a softmax function. Strengths: The advantage of the CNNs used here is the accuracy they manage to achieve in both approaches, an overall mean of 90% with the dice score component implemented and an 89.6% mean accuracy of combining aggregate predictions from the patches. The patches help decrease the sensitivity of image noise and imbalance backgrounds; robust design. Weakness: However, the model depends upon the balancing of the classes in the training set as this	CNN with images scaled to 60x60 px; first approach using 5-fold with 10 convolution layers with 3x3 filters, 2x2 max-pooling layers, a dense layer with 40 neurons using ReLU activation function, and a softmax output layer into 20 different classes. The second approach splitting the images into	The dataset comprised 1,769 images of Baltic amber gemstones from their collection at Kaunas University of Technology with RBG images of 640x480 px belonging to 20 classes.

	dataset used here was highly imbalanced, with some classes achieving lower accuracy than others. Other cases, do not have as high recall, and precision values as accuracy (not exactly shown in the paper, represented through the confusion matrix) since some color classes are extremely similar to each other and difficult to differentiate at times. The patch approach may improve the accuracy of the model, yet the increases in processing complexity do not appeal to our study.	60x60 patches and getting predictions based on the majority vote rule.	
[17] Deep Multi-feature Fusion CNNs with Gemstone Image Classification Algorithm (2022)	The model used here integrates deep multi-feature fusion for gemstone image classification, combining color and spatial features. First, the main color features are extracted using the k-means++ clustering algorithm. Simultaneously, spatial features are obtained via a denoising CNN. These features are fused in a multi-layered CNN architecture consisting of four convolutional layers with varying kernel sizes and pooling layers, followed by fully connected layers and a Softmax classifier for final predictions. Dropout is employed to mitigate overfitting, and the model is optimized using the Adam optimizer with a cross-entropy loss function. Strengths: The MCNN+ model achieves an accuracy of 82%, outperforming traditional CNN models by nearly 9% by combining color and spatial features, providing a more comprehensive understanding of gemstone images. Weakness: The Gemstones dataset having only 3,200 images with significant variability in lighting and background could affect model generalization.	-	Gemstone Images from Kaggle- 3200 images and 87 classes. With 2800 images for training and 400 for testing.
[18] Rotation Equivariance for Diamond Identification (2023)	This paper explores a Convolutional Neural Network (CNN)-based approach for diamond identification using a ResNet-18 model pre-trained on ImageNet. The model allows all weights to train and utilizes Stochastic Gradient Descent (SGD) with a learning rate of 0.1, momentum of 0.95, and no weight decay. The training process includes a linear learning rate warm-up for the first 400 iterations and learning rate decay at 6 and 9 epochs, with a maximum of 10 epochs. Images are preprocessed with resizing, random cropping, and normalization using either ImageNet statistics or dataset-specific	CNN with transfer learning via training on Resnet 18	The custom dataset comprises around 17,480 query images and 2,050 gallery images, with diamonds consistently captured in a specific scale

	statistics. A polar warping technique is applied to align diamond images around their centroids, improving scale invariance. During training, the model uses a fully connected layer for classification with softmax cross-entropy loss. The embeddings generated by the model are used to compute mean Average Precision (mAP) by comparing cosine similarities between query and gallery images. Ablation studies indicate the positive impact of polar warping and dataset-specific normalization on performance. Strengths: The model consistently achieved near-perfect mAP scores of up to 100%, showing high accuracy in diamond identification, even on unseen data. The custom GPU-based polar warping implementation significantly outperforms OpenCV's warpPolar, reducing computation time by up to 750 times, and making the process highly efficient for large datasets. Weakness: The model's reliance on fixed-scale images and polar warping may limit its usefulness in other datasets with varying scales or non-diamond objects. Incorporating polar warping increased training time by approximately 33% over baseline models, potentially affecting scalability in resource-constrained environments.		and orientation. Preprocessing involves image resizing, normalization, and polar transformation using diamond centroids.
[19] Deep Diamond Re-ID (2020)	The paper utilized CNN to recognize diamonds and re-identify them based on their imagery, as each diamond image would have a unique ID number attached to it. The Darknet19 CNN serves as the base architecture, and when the diamond goes past the test phase, it matches the original ID to validate the authenticity. The process of this CNN consists of the network declaring different classes with DarkNet-19 network and re-trained with ImageNet weights applied to each, all the layers captured, frame-by-frame; the network was fine-tuned with a dummy classification task on the dataset. The final fully connected layer is then replaced to fit the entire training dataset and the embedding process produces a vector to represent a unique ID for each diamond. The embedding process for the diamonds was compared using many different classifiers; KNNs (k = 1, 2,..., 5) and SVMs (N = 1,2,...,10).	DarkNet-19 CNN to serve as backbore architecture, and, KNN and SVM classifiers to recognize the augmented data	The dataset was set by the collaborators of the paper and collected numerous images of 100 different diamonds using ASET imaging.

	The optimal hyperparameter k-NN classifier with k = 5 and a N = 10 for reference images per diamond achieved a 100% mAP (mean Average Precision). Strengths: Their model has achieved a high accuracy with a validation of 99.7%. This model is not limited to just diamond entries and can be utilized for other applications. Weaknesses: The setup they have utilized for their training dataset is inaccessible and not easily portable to the public domain; the data have high dimensionality, and their dataset is limited as it is small compared to our project's initial dataset.		
[20] Sapphire Color Grading Method based on color Space Feature Automatic Clustering Algorithm (2024)	This paper proposes an automatic sapphire color classification method based on an unsupervised machine-learning algorithm that involves color space feature clustering. The process includes transforming the sapphire image from the RGB color space to a more suitable color space, such as CIELAB or HSV, followed by the extraction of color features like color histograms and moments. Clustering algorithms, including K-means and hierarchical clustering, are then used to classify sapphires with similar color features. The method's performance is compared with a neural network-based approach, showing higher accuracy and stability in color classification, particularly under various light sources and illumination conditions. Strengths: The proposed method shows high accuracy, with color classification exceeding 90%, and is stable under different lighting conditions. It adapts well to various color grading needs and offers significant improvements over neural network algorithms. Weaknesses: The automatic clustering algorithm relies on hand-crafted features like color histograms and moments, which may not be as adaptable to varying data or complex patterns. Additionally, they have not supplied details about the dataset they are using and hence we cannot conclude if the data they use is free of issues such as class imbalance .	The algorithm uses an unsupervised K-means and hierarchical clustering, for the classification task.	Presumably consists of sapphire images but no details were given
[21] Predicting knee osteoarthritis	The study utilized Elastic Net (EN) regression, Random Forests (RF), and Convolutional Neural	A CNN model, along	The study used data from

severity: comparative modeling based on patient's data and plain X-ray images (2019)	Networks (CNNs) to predict the severity of Knee Osteoarthritis (KOA). The EN model combined L1 and L2 penalties to shrink regression coefficients and selected variables with high predictive power. The RF model leveraged an ensemble of decision trees, using bagging and random variable selection to enhance prediction accuracy. A Linear Mixed Effect Model (LMM) was employed to account for the hierarchical structure of the data, incorporating a random effect to capture the correlation between knees within individuals. A CNN was used for predicting Knee Osteoarthritis (KOA) severity utilizing X-ray images from the Osteoarthritis Initiative (OAI) dataset. The CNN architecture included four convolutional layers, each followed by batch normalization and ReLU activation layers, and max pooling layers to reduce dimensionality. A fully connected layer and dropout layers were incorporated to prevent overfitting, with the network trained to minimize categorical cross-entropy loss using the Adam optimizer. The CNN model achieved a root mean square error (RMSE) of 0.77, demonstrating slightly higher prediction accuracy compared to Elastic Net and Random Forest models based on patient questionnaire data. The CNN model was designed to classify the severity of KOA according to the Kellgren-Lawrence (KL) grading scheme, leveraging advanced image processing techniques to capture relevant features from the knee X-rays. Strengths: The approach allowed for the identification of key variables that could be monitored for early interventions, and it provided comparable prediction accuracy between patient questionnaire data and X-ray images. Weaknesses: The accuracy of a model developed using patient assessment data is almost comparable to the CNN model. Moreover, the statistical models have an edge over the CNN model by identifying key variables that help the physicians to design interventions and help the patients for further treatment.	with Elastic Net Regression, Random Forests, and a Linear Mixed Effect Model (LLM) was used in the study.	the Osteoarthritis Initiative (OAI), which included baseline datasets of 4,796 individuals and specific knee X-ray images.
[22] A review of convolutional	Convolutional Neural Networks (CNNs) have revolutionized computer vision by enabling	None Applicable	None Applicable

neural networks in computer vision	effective hierarchical feature extraction through deep learning architectures. The paper provides an extensive review of CNN developments, highlighting their progression from early models like AlexNet to more advanced architectures such as ResNet, GoogLeNet, and Squeeze-and-Excitation Networks (SENet). AlexNet marked a significant milestone by introducing deep convolutional layers and ReLU activations, drastically improving image classification accuracy in large-scale datasets. ResNet introduced residual connections, addressing the vanishing gradient problem and allowing for deeper networks with better optimization. GoogLeNet's use of Inception modules provided computational efficiency by reducing parameter counts while maintaining high accuracy. SENet further enhanced CNNs by incorporating attention mechanisms that recalibrate channel-wise feature responses, emphasizing more relevant features for classification tasks. These architectural advancements demonstrate how CNNs have evolved to handle increasingly complex visual data while improving computational efficiency and model performance. The paper also delves into key challenges and future directions in CNN research. Overfitting, a persistent issue in deep learning, is addressed through regularization techniques such as dropout and batch normalization, which improve generalization by preventing over-reliance on specific features. Additionally, the paper highlights the importance of handling small object detection and fine-grained features using Feature Pyramid Networks (FPNs), which combine high-resolution and high-semantic information across scales. The need for efficient deployment on resource-constrained devices has led to the development of lightweight models like MobileNet, which use depthwise separable convolutions to reduce computational load while maintaining accuracy. Future research is expected to focus on integrating transformer-based models and self-attention mechanisms into CNN architectures, offering new possibilities for handling complex visual tasks with enhanced interpretability and efficiency. This comprehensive review underscores		

	CNNs' transformative role in computer vision and their potential for further innovations across diverse application domains.		
[23] Anomaly detection for synthetic diamond grain using monocular depth estimation (2024)	This paper studies the problem of estimating the maximum depth from a single 2D monocular image of a grain of synthetic diamond to detect anomalies that would breach the quality criteria. It proposes two models that employ different approaches to the depth map that combine the depth estimation model MiDaS and the EfficientNet image recognition model to detect anomalies in synthetic diamond grains. The first model, ERN, uses the normal map generated from the MiDaS-predicted depth map as an additional input to EfficientNet. The second part, ERP, applies pointwise convolution to the MiDaS-predicted depth map and uses it to weight the RGB image channels. It was tested extensively and the paper also used a 5x5 cross-validation method to estimate the error. ERP outperformed ERN, yielding an R^2 value of 0.853, an AUC of 0.966, and an F1 score of 0.731. Strengths: The proposed models outperformed state-of-the-art semi-supervised anomaly detection methods and a simple binary classification model in detecting anomalies in the synthetic diamond dataset. The models achieved high accuracy, with the ERP model having an R2 of 0.807 for maximum depth estimation. Weakness: The study is limited to a single dataset and did not demonstrate generalization to other datasets.	MiDaS (DNN) is combined with Efficient Net (CNN) through a concatenation-based model (ERN) and an attention-based model (ERP)	The created dataset consisted of 1,000 RGB-D images of synthetic diamond grains with a size of 256x256 pixels taken by a laser microscope from an overhead view of a single diamond.
[24] Faster R-CNN and DenseNet Regression for Glaucoma Detection in Retinal Fundus Images (2020)	The study proposes a two-step deep learning approach for glaucoma detection in retinal fundus images. The first stage employs Mask R-CNN, an advanced object detection network that extends Faster R-CNN by adding a segmentation branch. This network uses a backbone network (typically ResNet) for feature extraction, followed by a region proposal network (RPN) that identifies potential regions of interest. The RPN generates multiple potential regions, which are then refined using non-maximum suppression to eliminate redundant proposals. The network is trained using five loss functions: RPN class loss, RPN bounding box loss,	It employs a two-stage deep learning approach, utilizing Mask R-CNN with a ResNet backbone to precisely localize the optical disc in retinal fundus images, and, a	The study used two datasets: MESSIDOR (460 images) and Magrabi (94 images), both subsets of the RIGA dataset containing fundus images from different

3. Data Selection

The main goal of our project is to use images of round diamonds to extract the 4 Cs - Colour, Clarity, Cut, and Carat by passing them into a neural network that would give us the output. This, in turn, can be used as an input to a regression model using machine learning that would provide us the price from it. To fulfill this requirement we needed a dataset with images of diamonds. We decided to go with the Diamond Images Dataset on Kaggle. It contains images of diamonds scraped from the website of Capital WholeSale Diamonds. The images are situated in folders separated by the shape of the diamond, from which we choose 'Round' as it has the most amount of data and the most common diamond types found are round. Each image also has the following data associated with it within a CSV file:

- Path_to_img: Path to the images, allowing direct addition to a variable.
- Stock_number: Stock number with which each image file name corresponds,
- Shape: Shape of the diamond.
- Carat: Weight in carats of each diamond - one carat (ct) is equal to 200mg
- Clarity: Clarity index of each diamond. Figure 1 below shows the GIA clarity scale

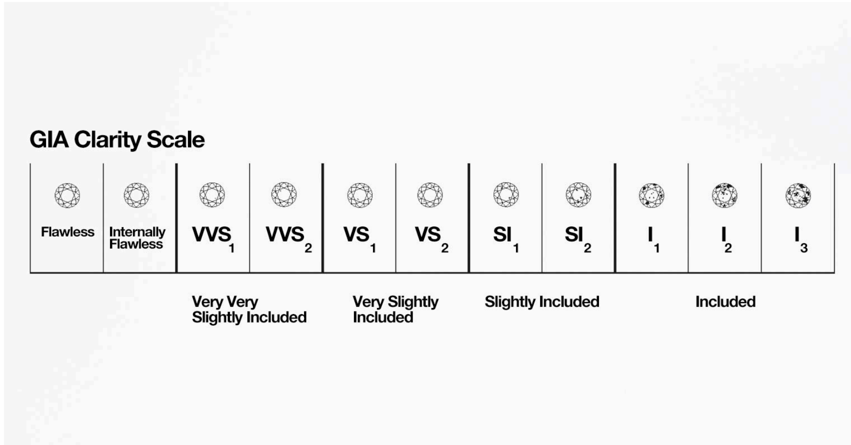


Figure 1: Clarity Scale

Color: Color index of each diamond. - A diamond is graded from D-Z based on its color. Figure 2 below shows the Color index.

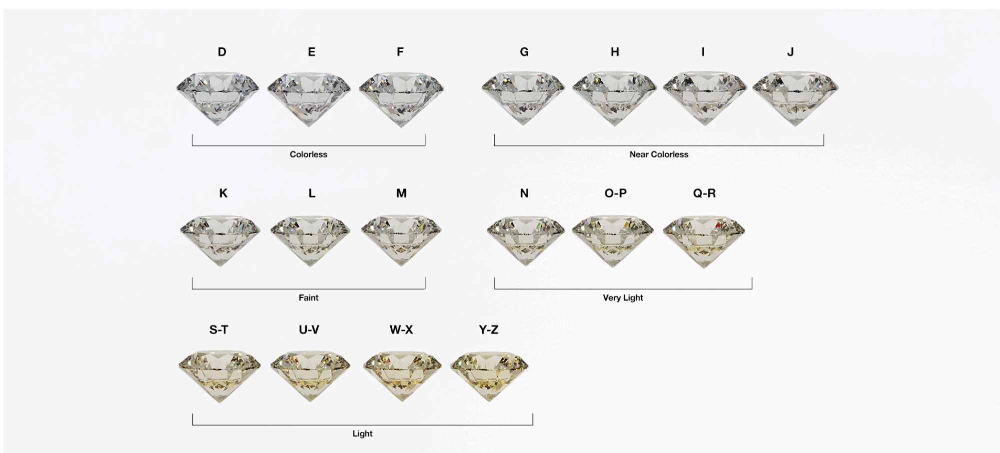


Figure 2: Color Index

- Cut: Cut grade of each Diamond. A diamond is graded from Excellent to Poor based on the quality of its cut. Figure 3 below shows the Cut Scale.

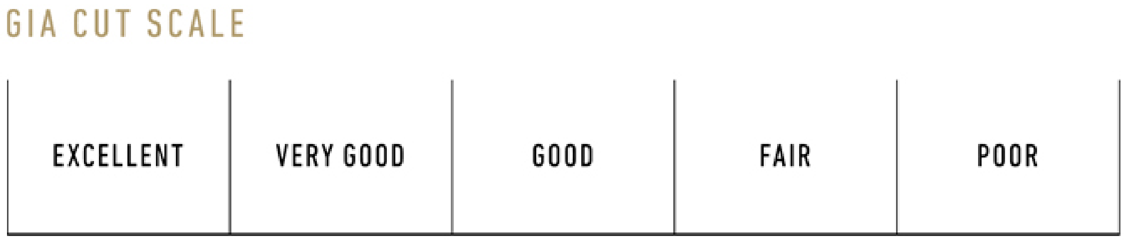


Figure 3: Cut Quality Scale

- Polish: Polish index of each diamond.
- Symmetry: Symmetry index of each diamond.
- Fluorescence: Fluorescence index of each diamond.
- Lab: Lab that certified the diamond
- Length, Width, and Depth of the diamond in millimeters.

The Round folder that we are using as our dataset consists of 21145 samples. Since the goal of our project is to determine the 4Cs, we dropped the extraneous columns and kept the 4 Cs: Cut, Clarity, Colour, and Carat. The image's dimensions are 600x474 pixels. We have chosen round diamonds because not only is the dataset dominated by round diamonds, but by specializing in round diamonds our neural network can focus on learning the features relevant to the 4Cs from the images. Since round diamonds also comprise the vast majority of diamonds traded in the market our model will also be applicable to real-life scenarios.

For the price prediction model, we used the ‘Diamonds’ dataset available on Kaggle since the ‘Diamond Images Dataset’ did not contain any price information. It has 53,000 samples, with the rows corresponding to the price in US Dollars, the carat weight of the diamond, quality of the cut, color, clarity, and carat along with the table, and x,y,z dimensions of the diamonds. Although the Data set at first appears to have good representation over the entire spectrum for carat, color, clarity, and cut, the dataset is highly imbalanced.

4. Data Cleaning and Feature Engineering

Step 1: Removing Rows with Missing Values To ensure the dataset's integrity, all rows containing missing values (NaN) in the critical columns—cut, clarity, and colour—were removed. This step ensures that only complete and meaningful data entries are used for training, eliminating potential issues arising from incomplete records.

Step 2: Removing Labels with Low Representation Labels in the columns (clarity, and colour) with fewer than 40 instances were identified and removed. This is because the vast majority of diamonds used in the industry fall in the ranges of D- M for colour and A - B for cut, . This threshold was set to eliminate under-represented classes that might skew the model's learning process or introduce bias. By focusing on adequately represented labels, this step improves dataset balance and ensures the robustness of the training process.

Image Preparation

Step 1: Copying Original Images All original images from the dataset, initially stored in the directory labeled round, were copied into a designated directory (augmented_round) for augmentation. This organization ensures that image processing tasks are streamlined, facilitating efficient augmentation and subsequent use in training.

Resizing Images: The images in the Kaggle dataset were originally sized at 600x474 pixels. These images were resized to 224x224 pixels to reduce computational overhead and expedite the training process. This resizing was designed to strike a balance between computational efficiency and the retention of key image features, ensuring that the resolution was sufficient to preserve the essential details necessary for accurate model training.

Threshold Calculation

Step 1: Recalculation of Augmentation Thresholds For each column representing the 4Cs (cut, clarity, colour), thresholds were computed to guide the augmentation process:

- **Over-Representation Threshold:** Labels with a frequency around 75% of the highest frequency labels were skipped during augmentation to avoid unnecessary duplication.

- **Under-Representation Target:** Labels with a frequency below 90% of the highest frequency were flagged for augmentation to achieve a more balanced distribution.

This dynamic thresholding mechanism ensures that over-represented labels are not further amplified while adequately augmenting under-represented ones to create a more equitable dataset.

Image Augmentation

Step 1: Augmenting Under-Represented Labels To address class imbalance, under-represented labels were augmented using the following transformations:

1. **Horizontal Flip:** A mirror image of the original was generated.
2. **Rotation:** Images were rotated randomly within ±20 degrees range to simulate orientation variance.
3. **Shift and Scale Adjustments:** Images were shifted and scaled by up to ±5% to mimic variations in size and positions.

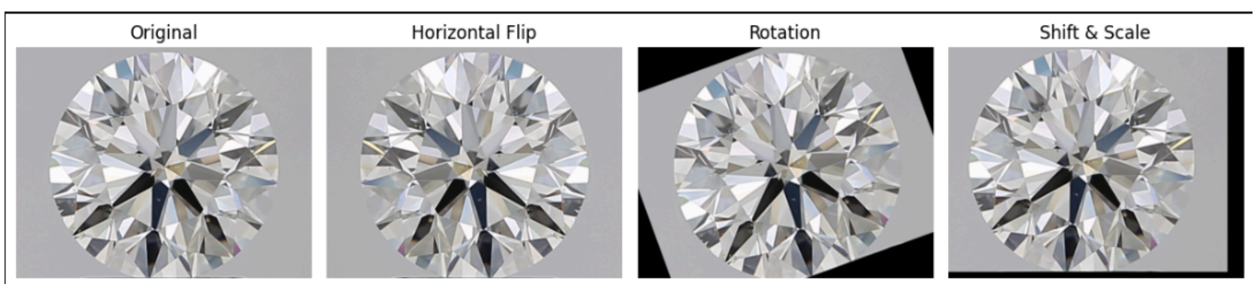


Figure 4: Diamond Image Augmentation

The model applied either a single augmentation or a combination of these transformations to each under-represented label's images. The augmentation process was designed to be minimal ensuring that the features of the images remained intact while enhancing data diversity. The augmented images along with the original images were then converted to a numpy array and then normalized for model learning purposes.

Dynamic Label Count Updates: The label counts were dynamically updated during the augmentation process, ensuring that no label exceeded its target frequency. This iterative adjustment maintained dataset balance and avoided the over-representation of augmented labels.

Data encoding: the compiled entries were then encoded ordinally using the GIA standard for better prediction. The color class was encoded into 23 categories using this standard similarly cut encoded into 5 categories and clarity into 8 categories. Although each category is distinct in terms of value the difference between each corresponding category in a class is small as even certified gemologists (Diamond graders) cannot discern these minute differences without specialized equipment. We grouped these different categories while maintaining a small amount of value variance inside each group for each class and got the following final encoding

Clarity Encoding	Name	Range	Number of Samples
1	Low	I1 - SI1	13993
2	Mid	SI2 - VVS2	24101
3	High	IF - FL	22346

Color Encoding	Name	Range	Number of Samples
1	Low	Y - Z to M	10894
2	Mid	L to H	17130
3	High	G to E	15027
4	Premium	D:P: BN to D	17389

Cut Encoding	Name	Range	Number of Samples
1	Low	Fair	265
2	Mid	Good to Very Good	24208
3	Premium	Excellent to Ideal	35967

As we can see from the above table, although the data is still imbalanced it is better than before. We also used sample weight to mitigate this imbalance as well. We used Class_weight ='balance' to get our sample_ weights for each of the three classes cut, clarity, and color.

```
Adjusted Class Weights:
Clarity Weights: {1: 1.4397675027990187, 2: 0.8359265867253087, 3: 0.901578209373788}
Color Weights: {1: 1.3870020194602533, 2: 0.8820782253356684, 3: 1.0055233912291208, 4: 0.8689401345678303}
Cut Weights (Adjusted): {1: 55, 2: 0.8322317691121393, 3: 0.5601430941325845}
```

Figure 4: Adjusted Class Weight

Finally, the dataset including the augmented, resized images was split using stratified splitting and saved in batches. As the model for price prediction uses the output directly from the classification of the main model therefore it uses the same encoding for its regression analysis. Shown below is the flow chart to show the overall flow of data processing and feature engineering.

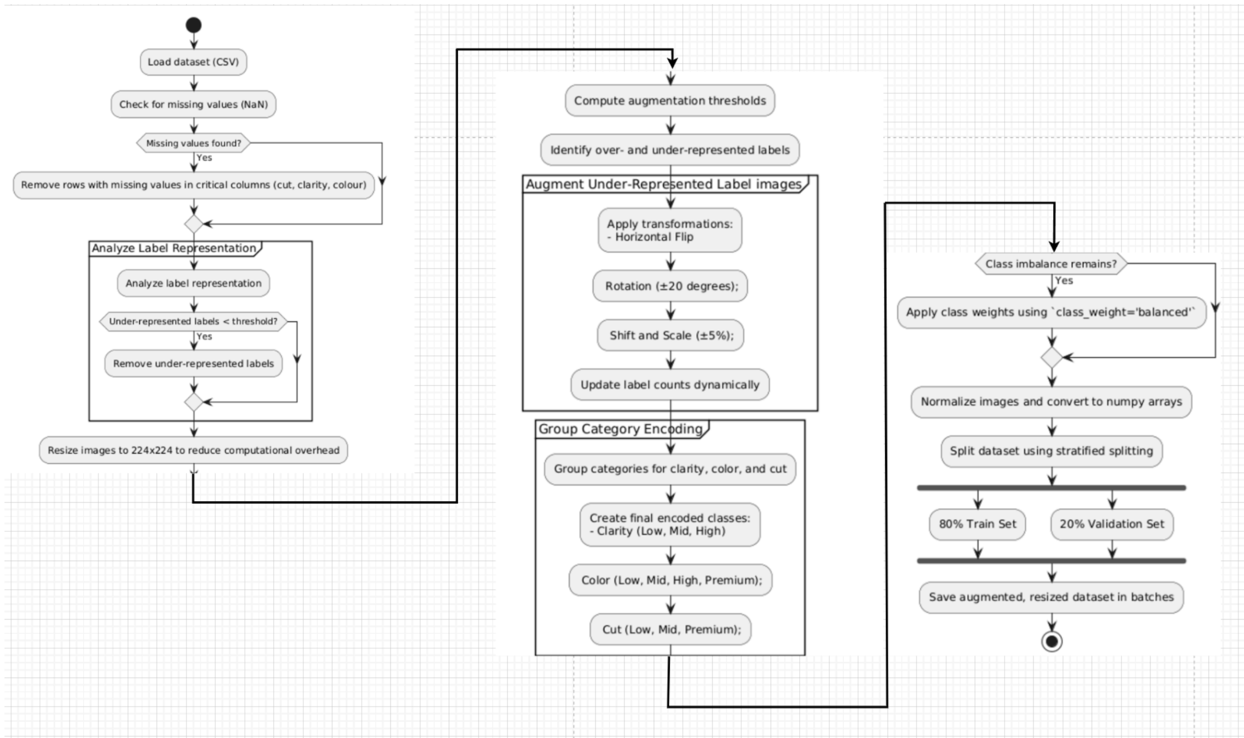


Figure 5: Data Processing Flow

This data preprocessing pipeline enabled efficient data utilization while ensuring that the dataset's quality and representativeness were optimized for training the model. Even though the dataset was highly imbalanced this approach made it more balanced across all the classification categories.

For the price prediction model, the Diamonds dataset was inspected for any missing values or empty columns, of which none were found. This was followed by dropping nonrelevant rows and encoding color, clarity, and cut to align with the encoding done for the image recognition model. Through this, we can directly use the predictions made by the image recognition model as the input for the price prediction.

5. Neural Network Architecture Selection

Model	Features
Multi-Output EfficientNet-B0	The first model we choose to attempt is a multi-output EfficientNet B0. Since our ideal target is a model that can run on smartphones, EfficientNet would be the best choice since it has better resource utilization due to its efficiency. Our sources also showcased high rates of success with EfficientNet with [7] obtaining an accuracy of 99.997% over other architectures such as AlexNet and Resnet, showing the high potential that EfficientNet can have for our problem. Moreover, EfficientNet was also used successfully in a related problem- Anomaly detection for synthetic diamond grain using monocular depth estimation [23] , showing that it can be useful for obtaining diamond features from its images. - We also used a shared conv2d layer for further feature extraction for shared meaning. - We then used simple 4 direct dense layers to get us four different outputs for the 4 C's of a diamond for our model.
Multi-Output 2-Dimensional Convolutional Neural Network	Our alternative model was crafting a 2 Dimensional Convolutional Neural Network. CNNs excel at image classification and regression tasks due to their ability to learn features step by step. The vast majority of papers we reviewed use CNNs for image classification, with Rotation Equivariance for Diamond Identification [18], Diamond Color Grading Based on Machine Vision [5], and Deep Diamond Re-ID [19]in particular showcasing the efficacy of 2D CNNs when applied to Diamonds. Building a 2D CNN can also help us obtain higher accuracies than using EfficientNet since we would be building the model from scratch and hence we would be able to customize its layers, batch size, and loss functions and optimize them to obtain high accuracies. - We used 4 direct dense branches connected to the main layers to get us four different outputs for the 4 C's of a diamond for our model. These branches used multiple dense layers along with batch normalization, L2 regularization, and dropout layers tuned specifically to each class

For the price prediction model, the top 3 models from [31], namely CatBoost, XGBoost, and RandomForest were selected for implementation along with a simple Linear Regression as the baseline for comparison purposes.

6. Why do chosen Neural Network Architectures Work

Model	Reasons for working
Multi-Output EfficientNet-B 0	Since we started this project with the end goal of developing a model usable in smartphones, and our dataset has around 60,000 images after augmentation, EfficientNet's compound scaling ensures optimal usage of computational resources, enabling it to handle large datasets effectively without compromising accuracy. Additionally, the task of classifying diamonds involves differentiating between small and subtle visual features related to cut, color, clarity, and carat. EfficientNet’s architecture, which uses depthwise separable convolutions, allows it to capture fine-grained features that are crucial for this type of classification.
Multi-Output 2-Dimensional CNN	Using multi-output 2D CNN for predicting the 4 Cs (cut, color, clarity, and carat) of diamonds from images would obtain us good accuracy metrics due to its ability to extract and learn visual patterns directly from raw image data. Convolutional layers excel at identifying spatial features such as edges, textures, reflections, and shapes, which are the visual elements that differentiate various diamond qualities. The cut of a diamond is obtained from its geometric structure and edge sharpness, while clarity involves detecting internal flaws and inclusions. By using a shared backbone of convolutional layers, the network efficiently captures these features, which are often relevant across multiple tasks. This shared learning process not only reduces redundancy but also enhances generalization, as the model leverages common visual cues to improve predictions for all four attributes. The multi-output design further optimizes performance by using specialized output layers tailored to each task. For categorical features cut, color, and clarity, softmax layers enable the model to classify images into distinct categories, while a linear activation layer is used for predicting the carat. In this way, the network can handle both classification and regression. Additionally, multi-task learning has reduced the risk of overfitting compared to training separate models for each attribute. Hence using 2D CNNs as the architecture streamlines the prediction process.

7. Validation Methodology

The validation process for this project was designed to rigorously evaluate the diamond classification model, but practical limitations shaped its execution. Initial data preprocessing involved removing missing values and applying ordinal encoding to clarity, cut, and color attributes to improve performance as well as reduce computational resources and time needed for training. The dataset was then split into training and testing sets using stratified splitting, with 80% allocated for training and 20% for testing. This split ensured balanced class distributions across subsets.

During training, the custom multi-output CNN model was trained on the augmented dataset, which was divided into 10 batches for efficient handling of computational resources. Training ran for 30 epochs to fine-tune model weights and each batch ran two times per epoch for better convergence. The team initially planned to implement advanced hyperparameter tuning techniques like GridSearch; however, this was found to be infeasible due to the complexity of handling the model's multi-output structure and resource constraints. Specifically, the alignment of multiple outputs (classification for the 4Cs and regression for carat) posed a significant challenge. GridSearch exacerbated computational demand, frequently causing crashes when executed in Colab. Custom hyperparameter tuning approaches were also considered but deemed too complex and beyond the project's scope due to time constraints.

Instead, the team adjusted parameters manually, iterating through values based on observed performance metrics. Although this approach was less exhaustive and more repetitive this allowed the team to stabilize the model within the available resources. Similarly, K-fold cross-validation was initially planned to enhance evaluation rigor but was ultimately not implemented due to the intricacies of managing the multi-output model's predictions and the significant computational load it required for the team's dataset and unique model. These complexities, combined with time constraints, led the team to rely on stratified splitting and batch-wise evaluations as an alternative.

The model's performance was assessed using metrics such as accuracy, precision, recall, F1 score, and ROC curves for the classification tasks, alongside MAE and MSE scores for the regression. Batch-wise evaluations revealed notable differences between the best- and worst-performing batches, highlighting potential inconsistencies in learning across subsets. This provided valuable insights into the model's behavior and areas for improvement.

Overall, while the lack of automated hyperparameter tuning and K-fold validation limited the scope of the evaluation, the project's validation process remained robust within its constraints. It highlighted key challenges, including the complexity of handling multi-output models and the need for extensive computational resources. The results emphasized the viability of the approach while pointing to future refinements, such as implementing more advanced tuning techniques and expanding computational capacity, to further enhance model performance and generalizability. The flowchart given below shows the overall validation process the team used for the image recognition model.

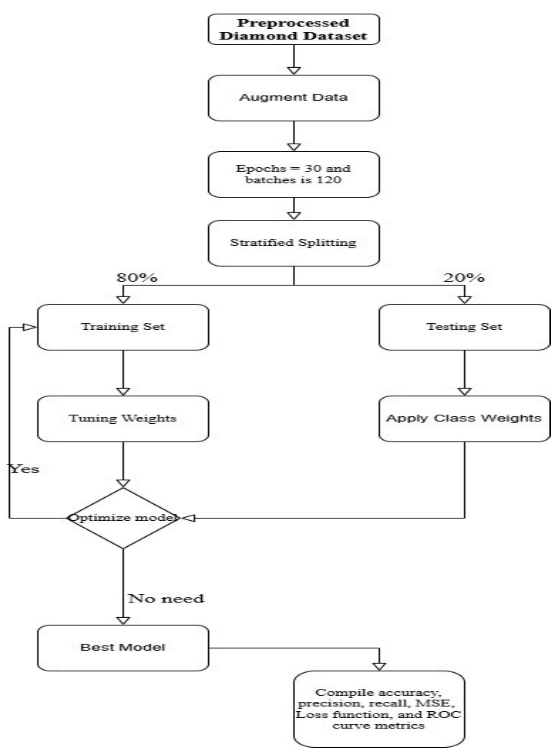


Figure 6: Validation Methodology

8. Results

EfficientNetB0 transfer learning model:

EfficientNetB0 was the first primary model tested during the project. EfficientNetB0 is a widely recognized and versatile pre-trained model known for its efficiency and effectiveness in transfer learning tasks. However, despite its established performance in other contexts, it fell short of expectations in this application. Even with its best-performing batch in batch (batch 9), it got an Average Accuracy of just 35.02% across all the four c's of the diamond and an Average Loss of 0.9553 suggesting that the overall model was unable to interpret critical information.

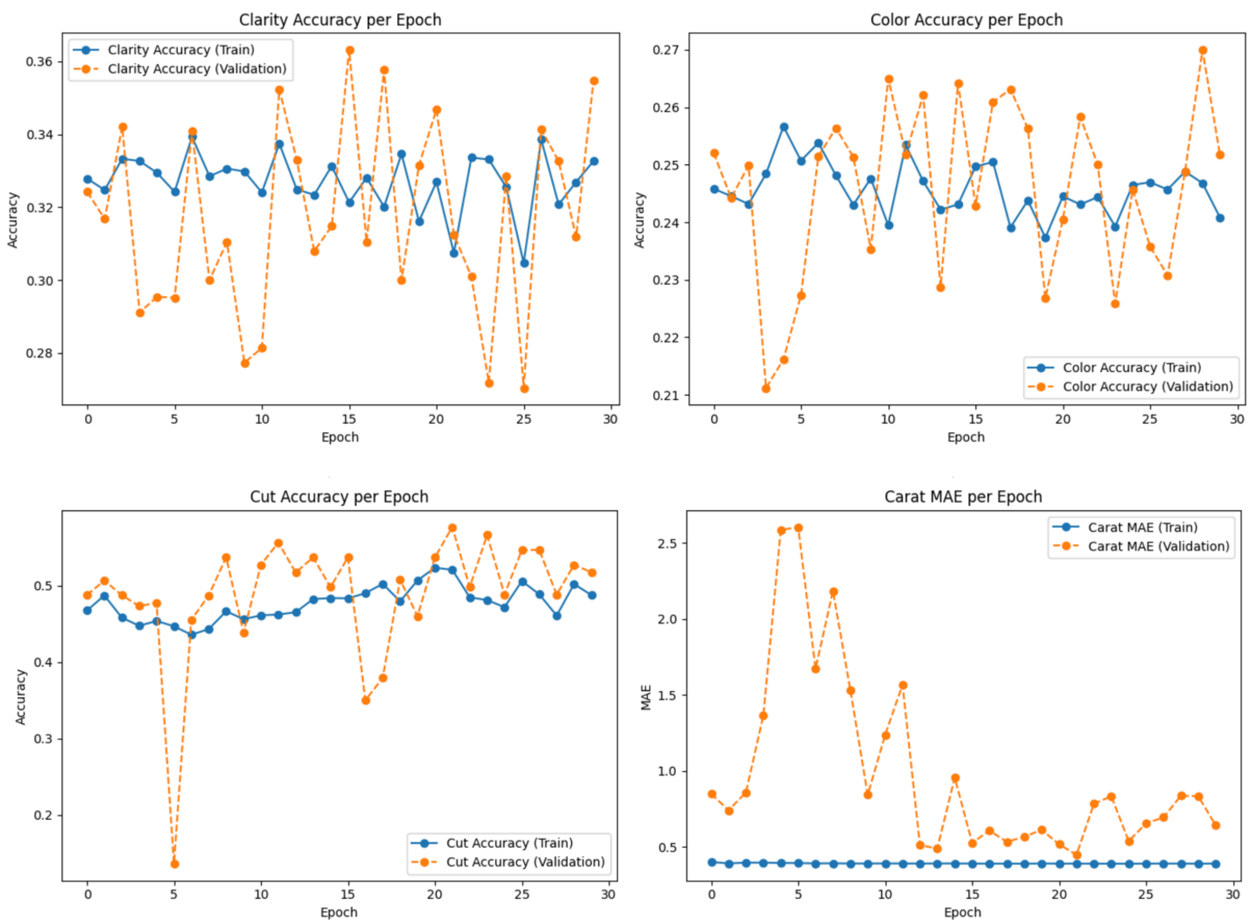


Figure 7: Training and Validation Accuracy per Epoch

The above three graphs other than Carat MAE show fluctuations without a clear upward trend that shows the model is unable to handle the imbalance in the data set. The overall accuracy could also be affected by the group encoding which will be further explained for the next model. Carat MAE exhibiting a clear downward trend shows us that the model can handle the regression part of the model better. The validation also clarifies that there is no overfitting or underfitting. The figures below are the confusion matrices for each of the three classifications.

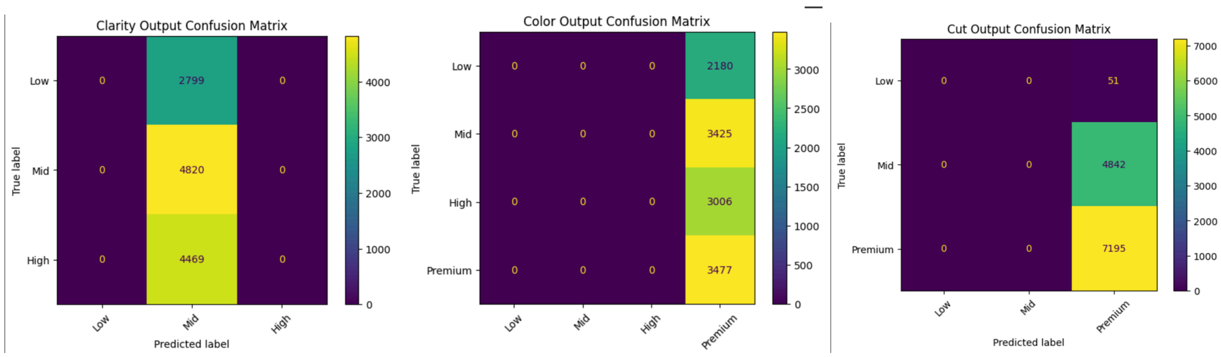


Figure 8: Confusion Matrices for EfficientNet

The confusion model shows us that the model is not predicting properly and only classifies each class with one label giving us the final table for f1 score, precision and recall.

Metric	Clarity (Avg	Color (Avg)	Cut (Avg)
Precision	0.13	0.07	0.2
Recall	0.33	0.25	0.33
F1-Score	0.19	0.11	0.25

This is very inaccurate. Due to this poor accuracy, the team decided to shift to making the custom multi-output CNN model proposed during the research phase as it could be trained from scratch to possibly increase the model's generalization and increase overall accuracy.

Proprietary Custom multi-output CNN model

We went through multiple iterations to make sure the inputs for the model were optimized. For the first iteration; the model used only normal ordinal encoding with no augmentation to images to reduce the imbalance. The below four graphs show the accuracy and loss for the training and validation data set for each of the four classes.

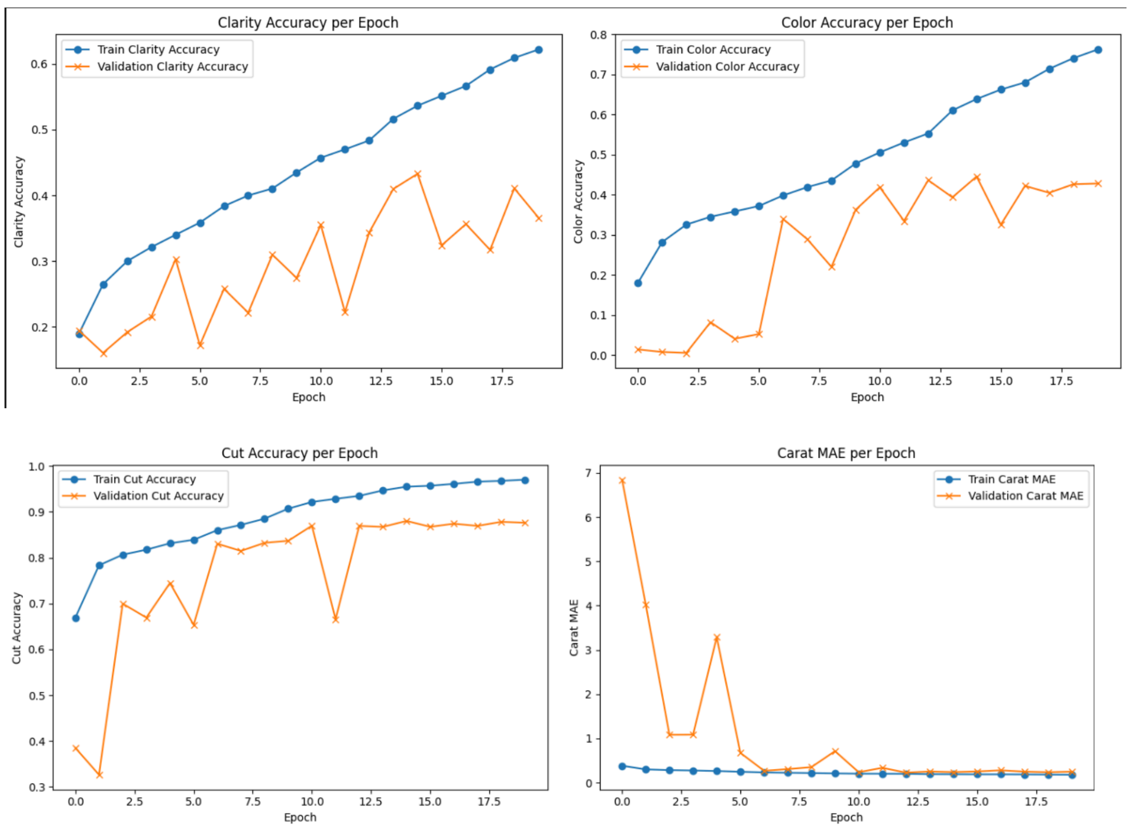


Figure 9: Training and Validation Accuracy per Epoch

It is clear from the above graphs the model seems to be learning better than before but struggles with overfitting with clarity and color branches. The below four graphs show the losses for each class. The loss curve also shows that the model is learning well with lower loss for both data sets (note: this is the only mode that used the image generator method for loading data which was later scrapped. While being better for memory usage, it produced a new set of issues to track results.)

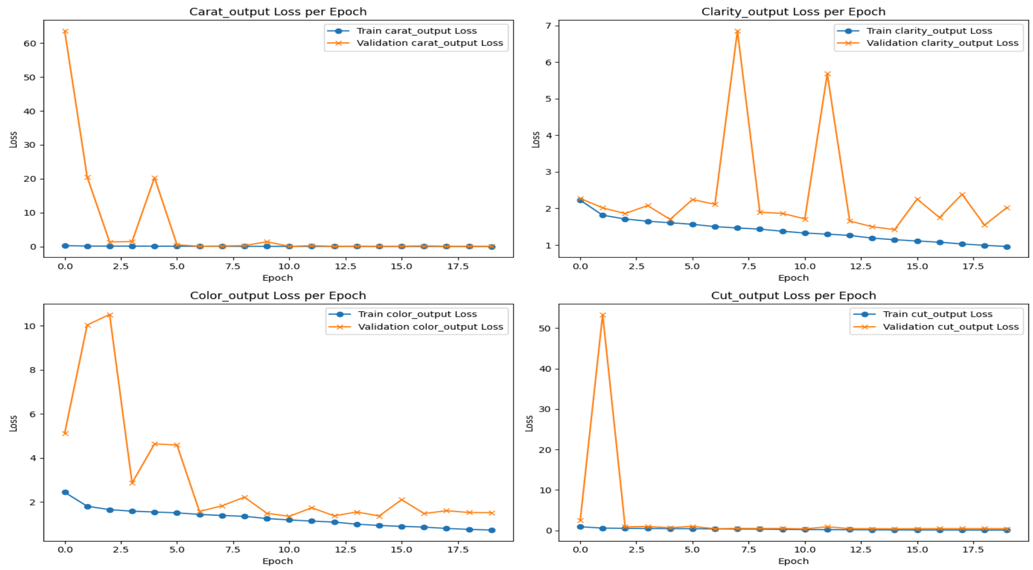


Figure 10: Training and Validation Accuracy per Epoch after changes

With the following results at the last epoch:

Category	Learn (Train)	Validation (Val)
Carat Output Loss	0.0689	0.1448
Carat Output MAE	0.1798	0.2483
Clarity Output Accuracy	0.6209	0.3660
Clarity Output Loss	0.9665	2.0260
Color Output Accuracy	0.7629	0.4282
Color Output Loss	0.7198	1.5104
Cut Output Accuracy	0.9699	0.8763
Cut Output Loss	0.0935	0.4456
Total Loss	**2.1523**	**4.4314**

The model's overall performance proved to be more promising than the previous model therefore the team thought it would be better to enhance the data augmentation and balancing to reduce overfitting to achieve high accuracy. The first step the team took to improve was to augment the image dataset using shift rotate and so on as explained in the preprocessing section. This reduced the data imbalance to a significant degree but the data was still very skewed. Hence we thought of grouping labels using the GIA standard [31] to maintain data balance and avoid making the grouping vague. Grouping was done manually to ensure it met the above requirements. The team made two sets of group encoding to see the difference in the model's performance.

Group encoding :

Set1 data balance:

Clarity Rank	Group	Clarity Count
3	High Clarity	1396
2	Mid-High Clarity	11556
1	Mid Clarity	9394
Color Rank		Color Count
5	Rare Color	175
4	Premium Color	5754
3	High Color	26487
2	Mid Color	24475
1	Low Color	3549
Cut Rank	Group	Cut Count
3	Premium Cut	55503
2	Mid Cut	265
1	Low Cut	4492

Set2 data balance:

Clarity Rank	Group	Clarity Count
3	High Clarity	22346
2	Mid-High Clarity	24101
1	Mid Clarity	13993
Color Rank	Group	Color Count
4	Premium Color	17389
3	High Color	15027
2	Mid Color	17130
1	Low Color	10894
Cut Rank	Group	Cut Count
3	Premium Cut	35967
2	Mid Cut	24208
1	Low Cut	265

Grouping was a practical approach to address imbalances in the dataset, especially when certain classes, such as Level 15 clarity even after augmentation had only 24 entries. By combining such levels into broader categories, the model avoids skewed performance and ensures sufficient data in each group for meaningful learning. Grouping also enhances generalization, as diamonds with overlapping characteristics or similar pricing tiers can be better represented through simplified categories. Additionally, this method improves interpretability, aligning the dataset with real-world terms like "premium cut" or "high clarity," making predictions more intuitive for consumers with the tradeoff of being less effective for professional graders.

Grouping positively affected the model by reducing noise from underrepresented classes, enabling it to focus on balanced patterns, and simplifying decision boundaries for more accurate and generalized predictions. However, it trades off fine-grained distinctions, sacrificing detail for simplicity. For pricing, grouping tends to mask small variations within adjacent levels but captures broader trends effectively. For instance, High Clarity and Premium Color groups generally command higher prices, while Premium Cuts significantly differ from Low Cuts in value. Variations in mid-to-low color levels may not substantially impact pricing, making grouping a practical choice for broader pricing tiers.

The team then ran two more tests with both groups encoding to see the difference in classification accuracy.

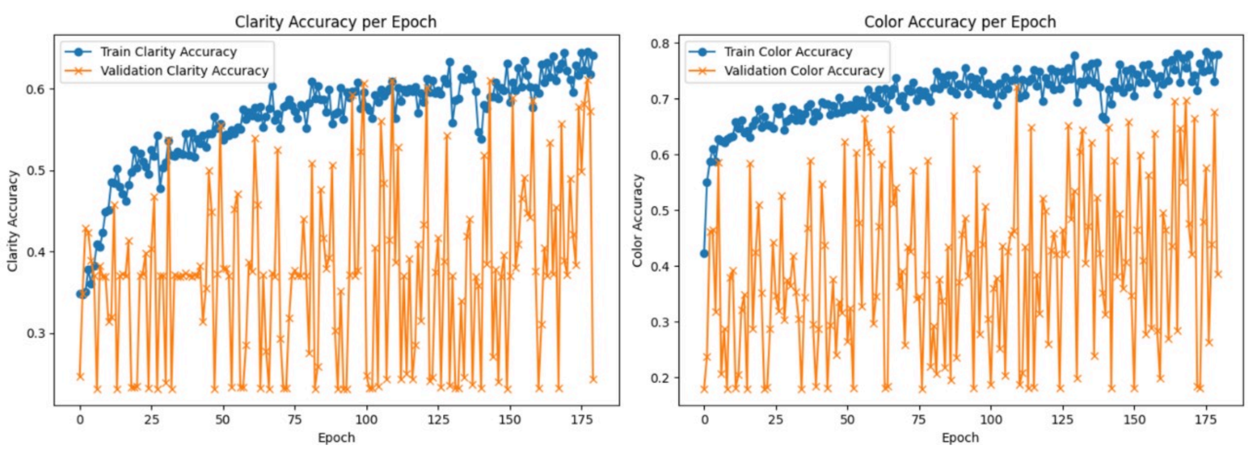


Figure 11: Training and Validation for Test3 using set1

The model from the test3 onwards stratified split the dataset into 10 .pkl batch files for training and three for validation this was done to reduce memory load but this also led to each batch reacting differently to the model training which is why there is a great discrepancy in this graph and not due to improper learning. The team also used sample weights for each label for the different classes to reduce the dataset imbalance from the above graph we see that the model even though it had a few phases where the accuracy plateaued the overall accuracy increased for all four classifications.

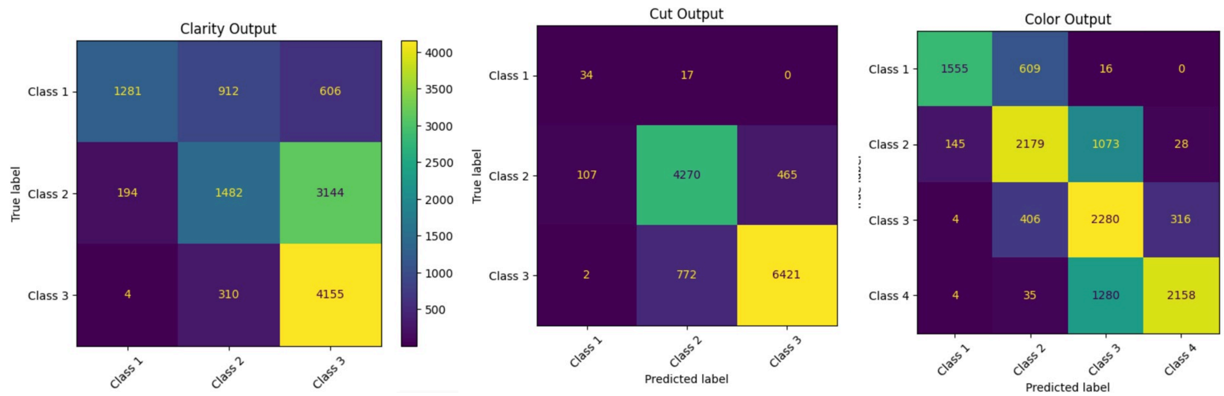


Figure 12: Classification Matrices

We see that from the above three matrices, there is a better overall performance of the model with higher precision and recall for cut and color although clarity is lacking behind the overall performance is way better. The overview of the average performance for each class for all its labels is shown below.

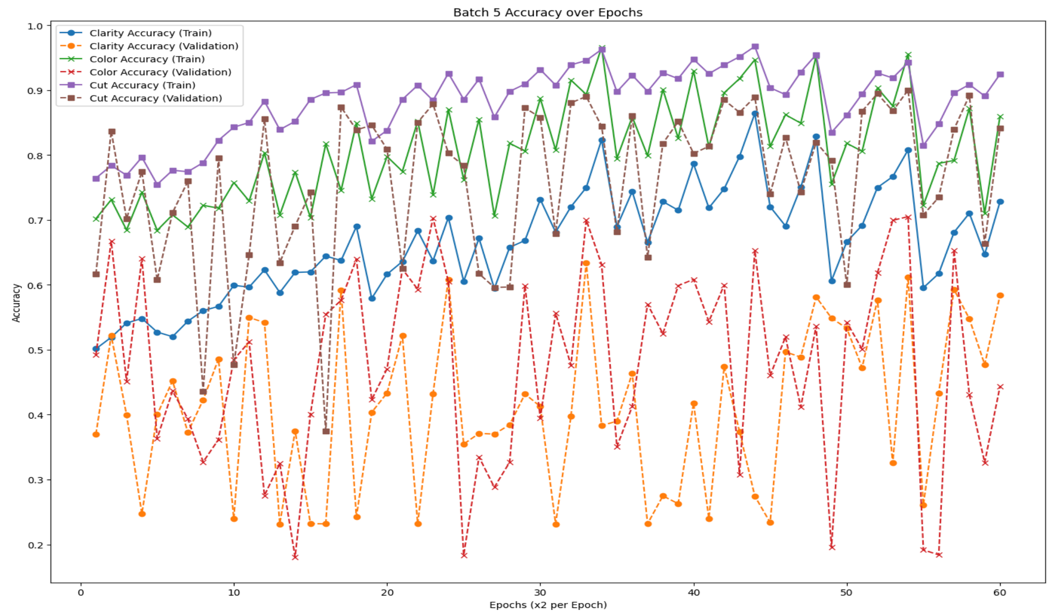
Metric	Clarity	Cut	Color
Precision (Avg)	0.65	0.67	0.73
Recall (Avg)	0.56	0.81	0.68
F1-Score (Avg)	0.55	0.71	0.69

We also observed that very minimal to no extreme classification shows that the model understands most of the unique properties of each label in the class but was not able to distinguish minute differences even though they are different diamonds due to noise and actual difference being low between each class of diamonds the model tends to favor the slightly higher /lower grade as they appear similar to the actual grade.

Final Test (No 4) set2:

To better evaluate our final model we went into more detail to see how each batch reacted to the learning process then compiled this data for each epoch to get a better understanding of the model evaluation. We can see that for the Best Performing batch, Batch 10 we got Average

Accuracy: 0.8003, Average Loss: 0.3535, and for our Worst Performing batch, Batch 4 - Average Accuracy: 0.7832, Average Loss: 0.3776.



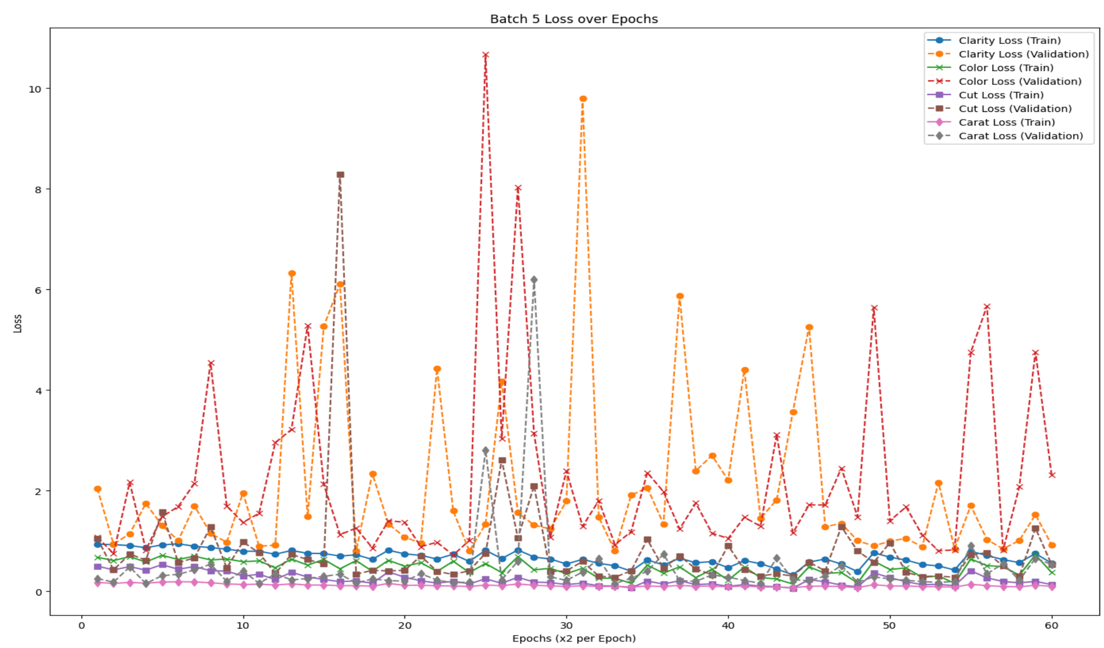


Figure 13: Batch 5 Training and Testing Loss over Epochs

The above graphs show the accuracy and loss curve of Batch 5 giving us a better understanding of the model's behavior with different batches.

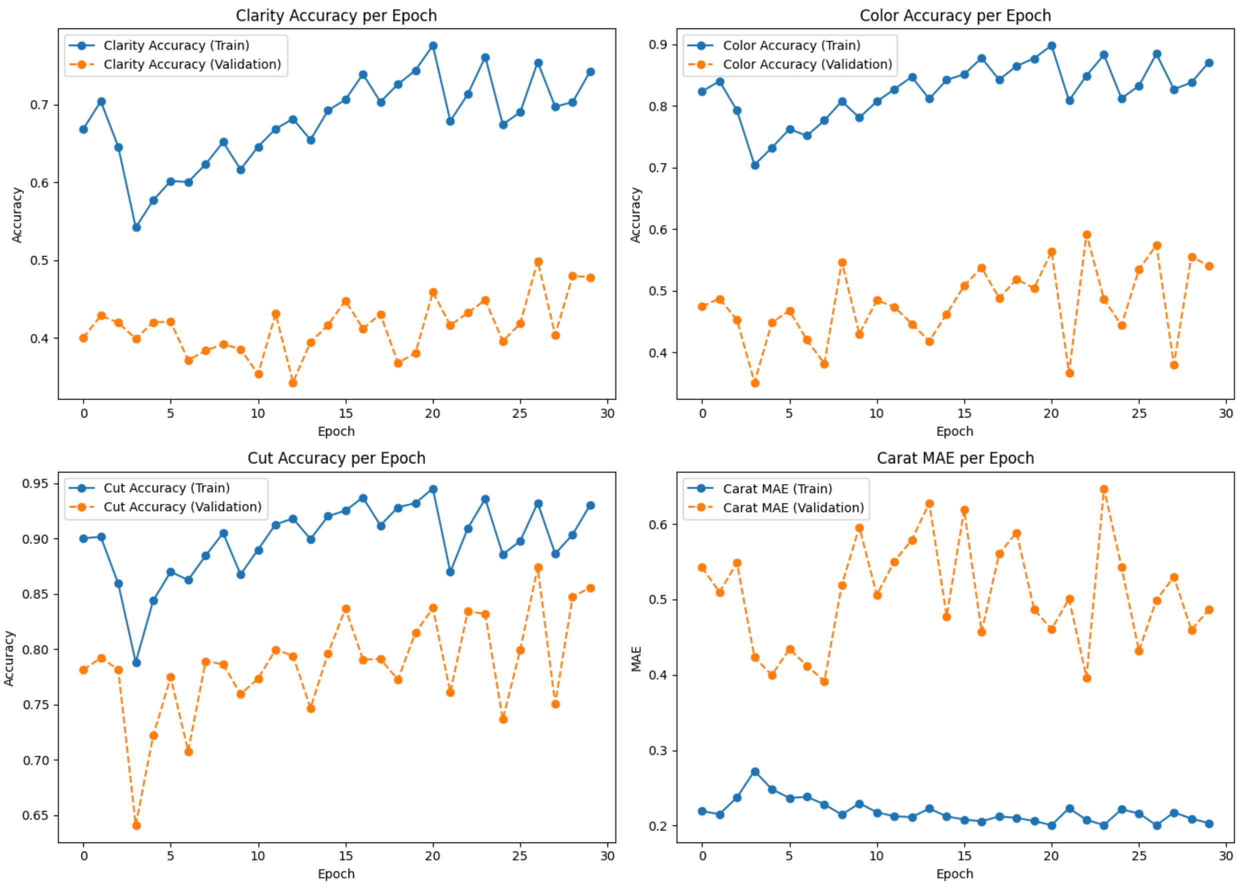


Figure 14: Training and Testing Accuracy per Epoch

The model shows signs of overfitting still even after applying multiple data augmentation techniques but due to the nature of the dataset this is well within expectations. The data was initially very skewed and even after augmentation it still retains a lot of the inherent imbalance. Although the gap between the validation and training curves may seem high, both still appear to follow the general trend. We also see a similar trend in the losses as seen below.

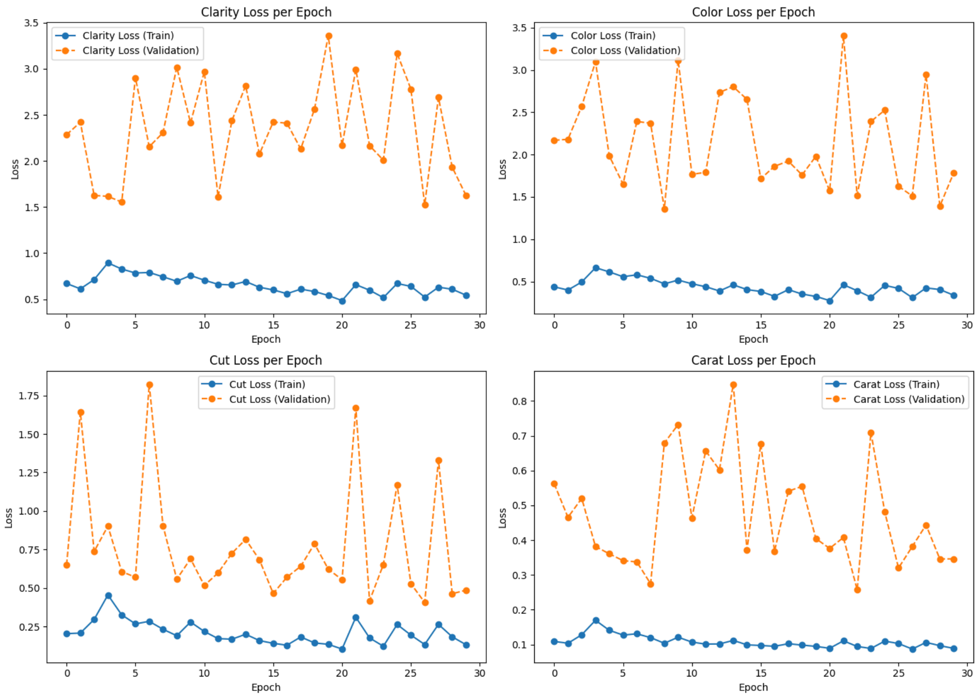


Figure 15: Final Training and Testing Accuracy per Epoch

The below confusion matrix also shows improvement in the accuracy of the mode due to set 2 group encoding but due to time constraints, the team could not fine-tune this parameter even further for better accuracy metrics.

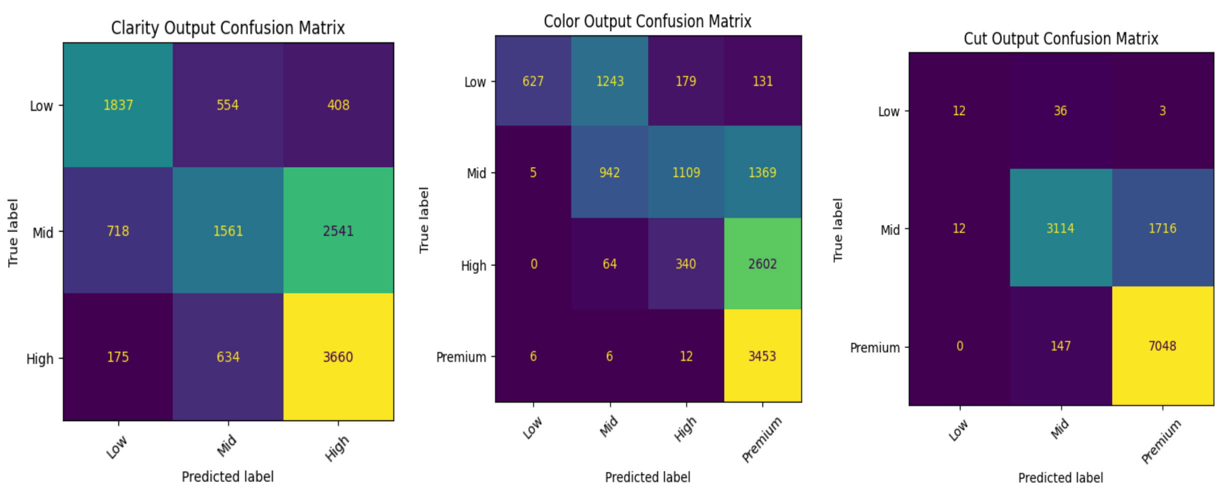


Figure 16: Final Training and Testing Accuracy per Epoch

Here is a detailed breakdown of our results for the final model

Cut Metric			
Label	Precision	Recall	F1-Score
Class 1	0.5	0.235	0.32
Class 2	0.944	0.643	0.765
Class 3	0.804	0.98	0.883
Average	0.749	0.619	0.656

Color Metric			
Label	Precision	Recall	F1-Score
Class 1	0.983	0.288	0.445
Class 2	0.417	0.275	0.331
Class 3	0.207	0.113	0.146
Class 4	0.457	0.991	0.626
Average	0.516	0.417	0.387

Clarity Metric			
Label	Precision	Recall	F1-Score
Class 1	0.673	0.656	0.664
Class 2	0.568	0.324	0.412
Class 3	0.554	0.819	0.661
Average	0.598	0.6	0.579

For the Carat regression model, we got MAE (Average): 0.223 and Loss (Average): 0.113 accuracy metrics. This means that, on average, the model's prediction for the carat weight is off by 0.223 units of carat. The lower MAE values indicate better overall performance of the model. Lower loss values indicate that the model is a better fit for the data for this aspect of the diamond. The overall results show that the model can predict the four C's of a diamond within acceptable accuracy through its novel approach to classify the diamond using a multi-class multi-output 2d CNN approach.

ROC Curves and Analysis

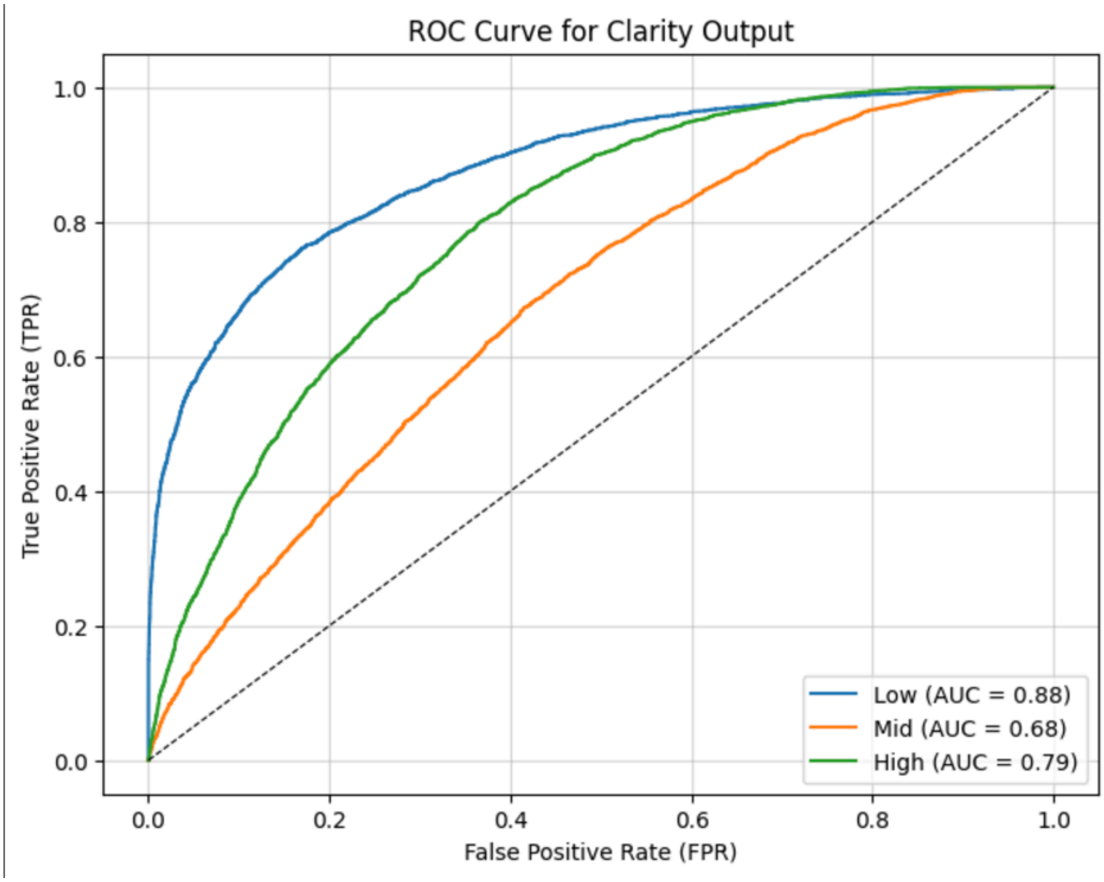


Figure 17: Roc Curve for Clarity

The ROC curve for the Clarity output shows varying performance across the three categories. The Low-clarity category achieves an AUC of 0.88, indicating a strong ability to distinguish between low-clarity diamonds and others. This suggests that the model is performing well for this class and can reliably detect low-clarity instances. The Mid clarity category has an AUC of 0.68, which indicates moderate performance. While it shows some predictive power, it's not particularly strong and could benefit from further refinement. Lastly, the High clarity category achieves an AUC of 0.79, which is good considering it is the first attempt at the neural network that predicts diamond clarity from normal images but still leaves room for improvement. Overall, the model performs well for the Low clarity class, and High is good, Medium could use further refinement.

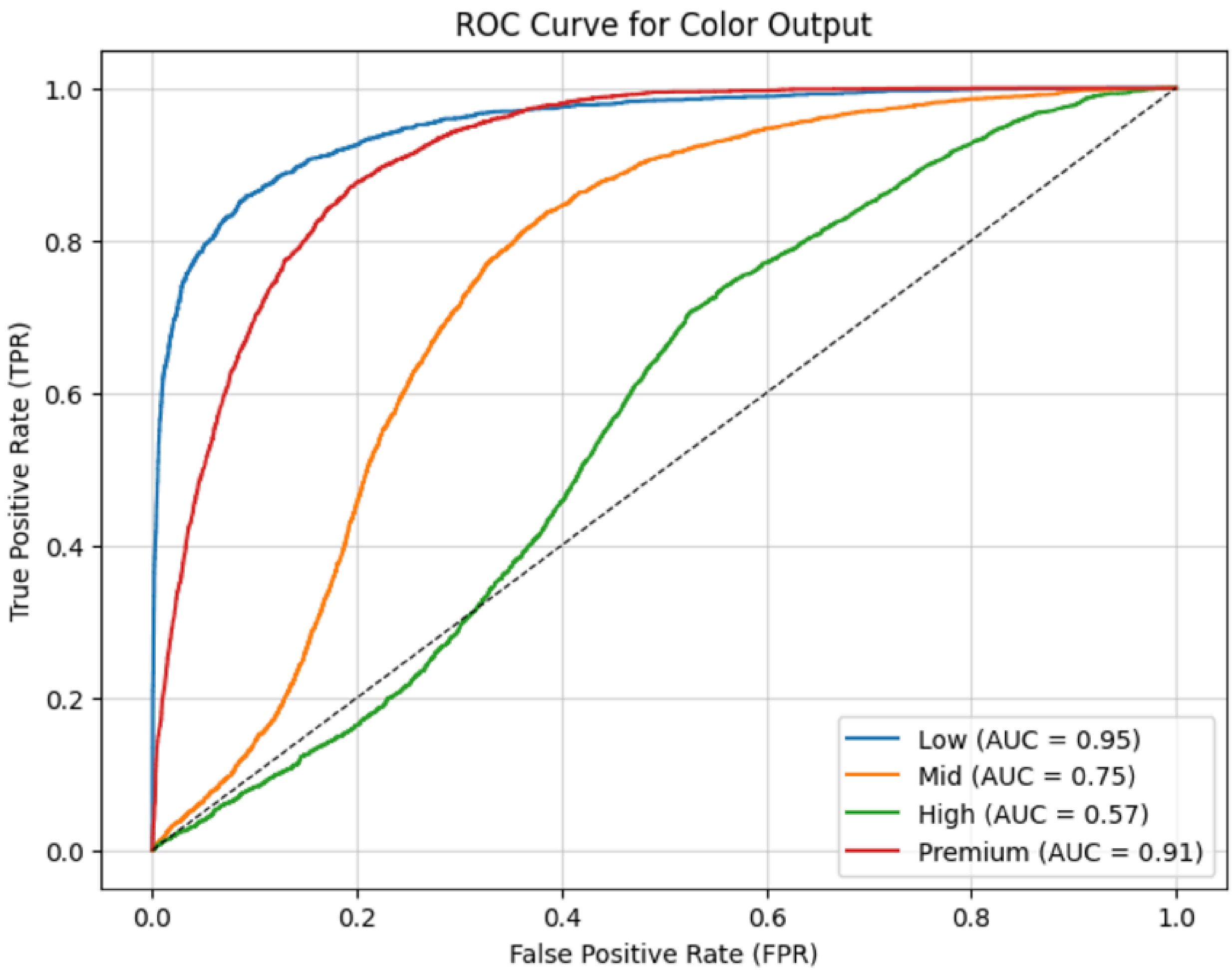


Figure 18: Curve Curve for Clarity

The ROC curve for the Color output demonstrates similarly mixed results across the different categories. The Low color category stands out with an impressive AUC of 0.95, indicating that the model can effectively distinguish low-color diamonds from other grades, providing excellent classification performance. Similarly, the Premium color category also performs well, with an AUC of 0.91, showing strong predictive capability. However, the Mid color category has an AUC of 0.75, suggesting moderate performance with room for improvement. The most concerning result is for the High color category, which has an AUC of 0.57, and thus needs a lot more work. This indicates that the model struggles significantly with distinguishing high-color diamonds, likely due to overlapping visual features with premium and medium. Improving performance for the High and Mid categories would involve addressing these issues through enhanced image preprocessing, or model tuning.

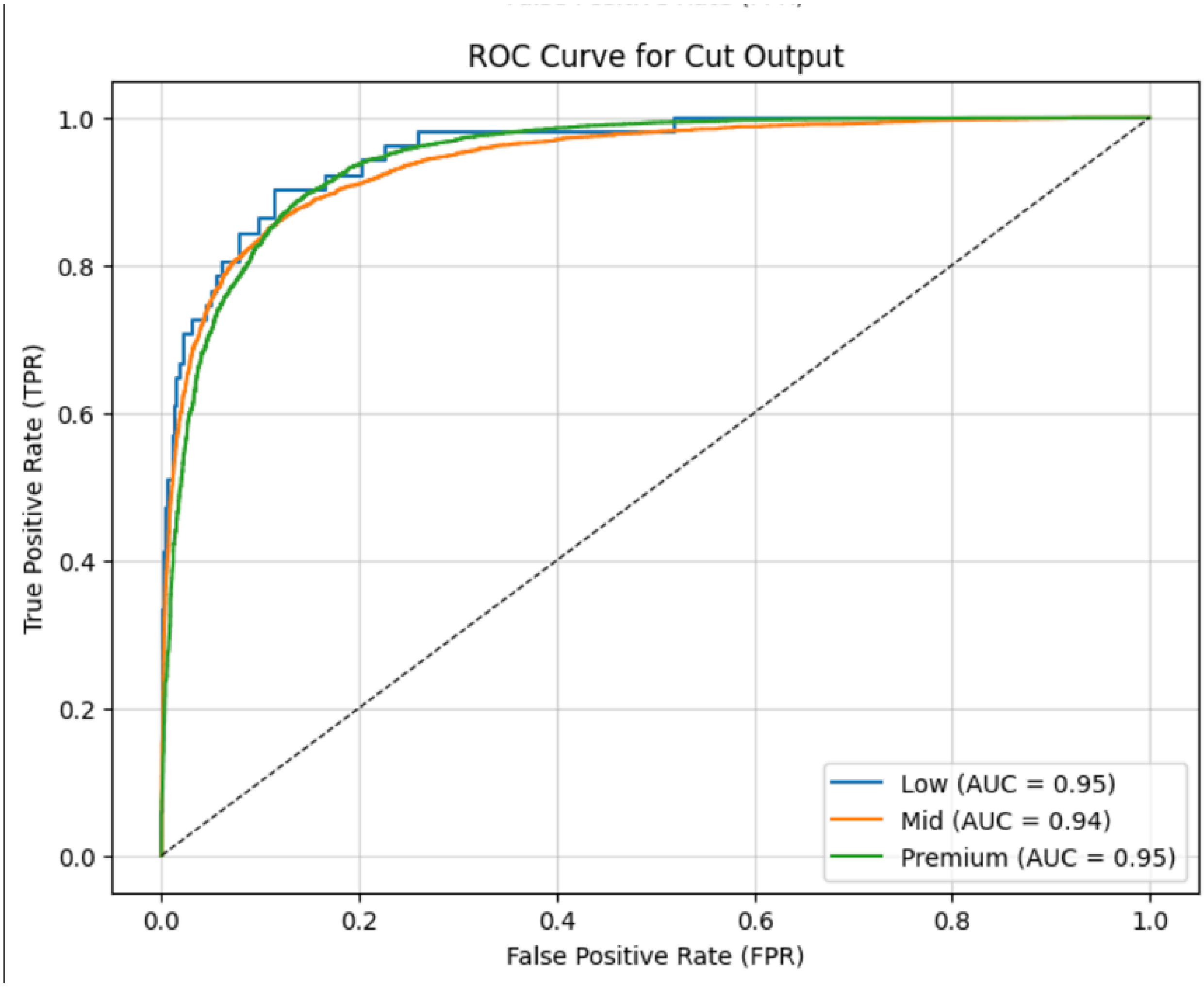


Figure 19: Roc Curve for Cut

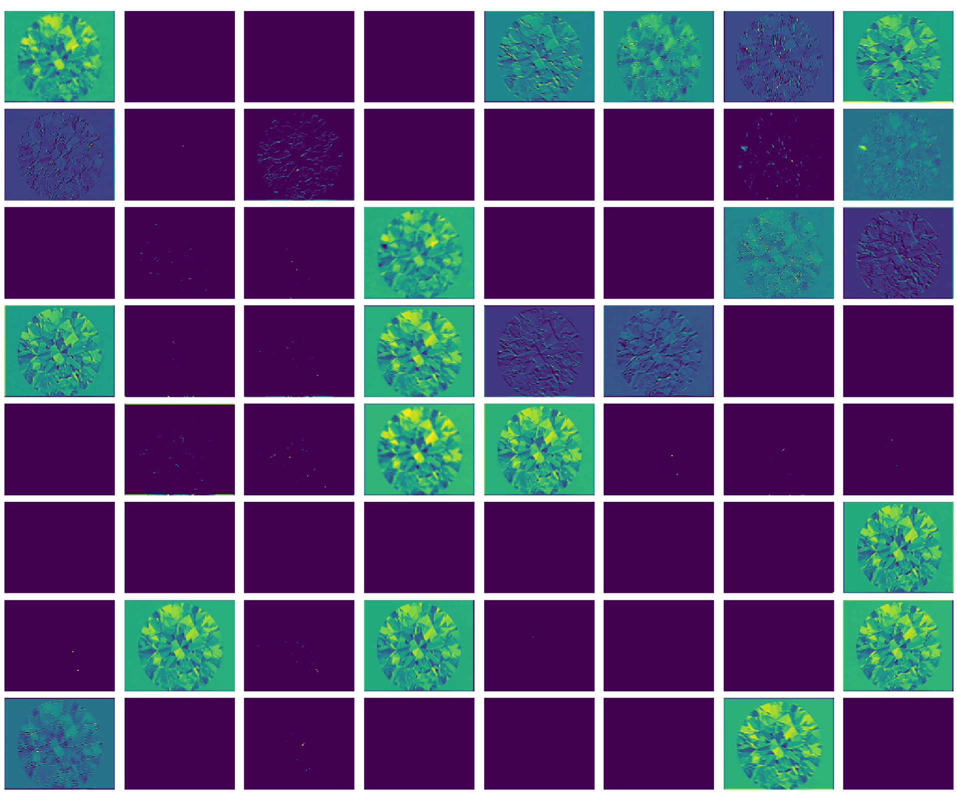
The ROC curve for the Cut output shows consistently strong performance across all categories. The Low and Premium cut categories both achieve an AUC of 0.95, indicating excellent classification accuracy. The Mid cut category follows closely with an AUC of 0.94, reflecting similarly high performance. These results suggest the model effectively distinguishes between different cut levels, likely due to well-balanced data and effective feature extraction. With all AUC values above 0.94, the model demonstrates a high level of reliability in predicting diamond cuts, requiring minimal further improvement.

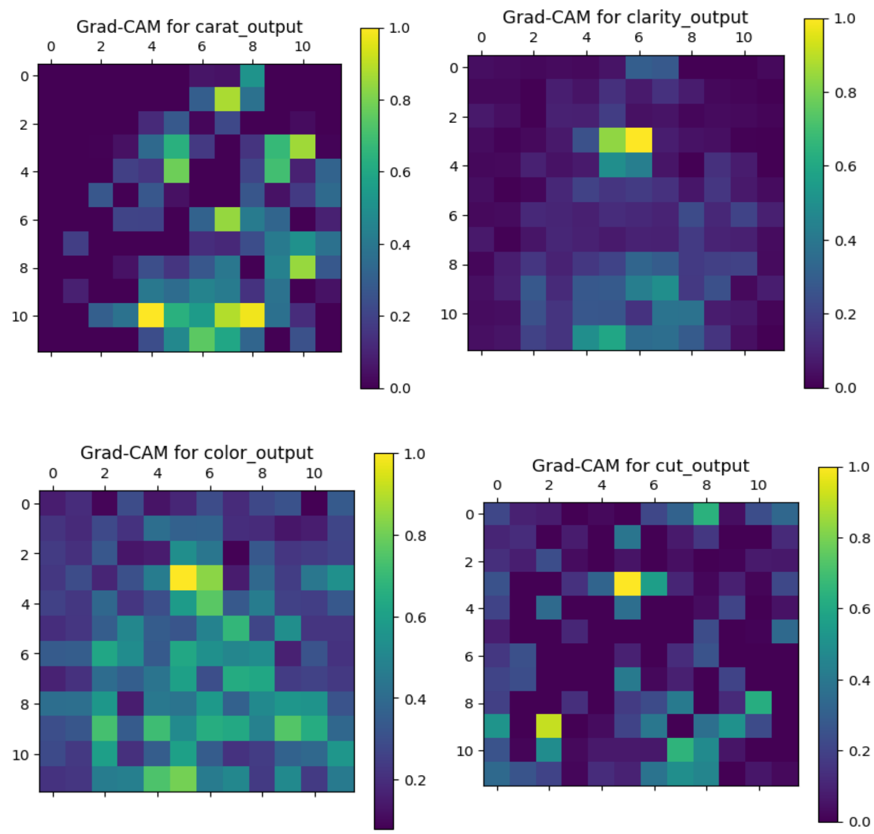
The team also wanted to visualize how the diamond was being processed. Hence feature maps and GRAD-CAM were used to achieve visualization. Gradient-weighted Class Activation Mapping (GRAD-CAM) is a visualization technique used to understand the decisions made by deep neural networks, particularly convolutional neural networks (CNNs). It highlights the regions in an input image that are most relevant to a specific prediction made by the model. These gradients indicate how important each feature map is for the model's prediction.

Figure 20-25 Feature Maps and Grad-Cams for model

Filters from Layer conv2d

Feature Maps from Layer conv2d





Price Prediction

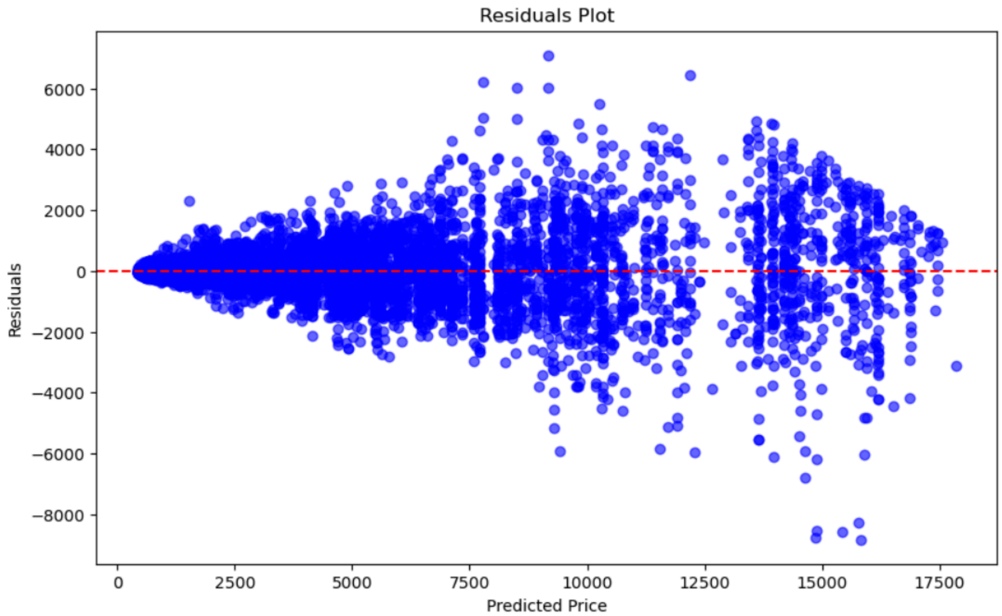
For the price prediction model, the data was split into a random 70:30 train-test split. Then the selected models- XGBoost, CatBoost, and RandomForest were trained along with Linear Regression as the basis for comparison. The following accuracy metrics (Mean Absolute Error, Root Mean Square Error, and R^2 were obtained:

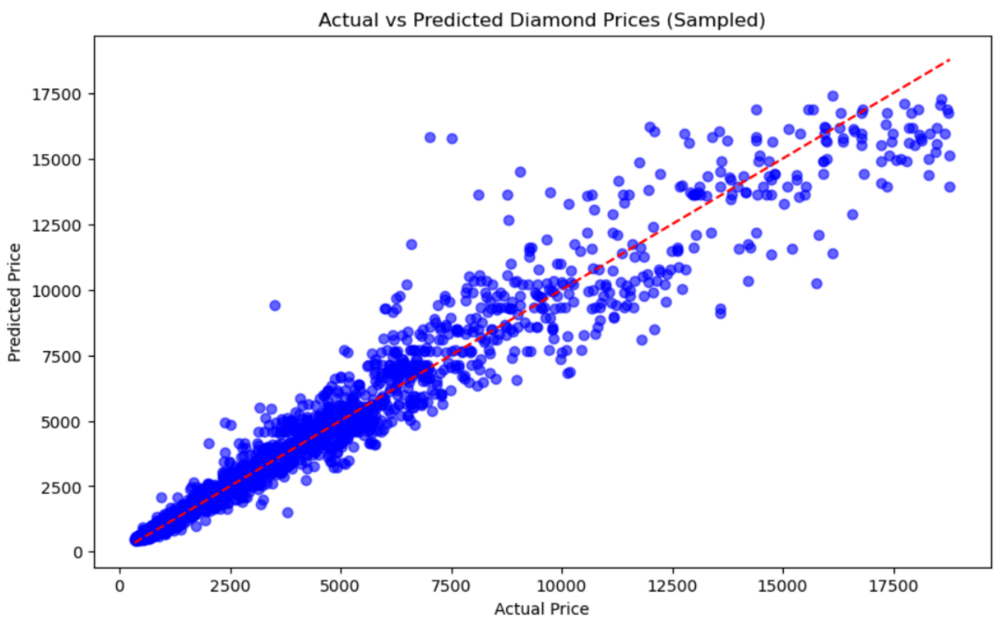
	Model	Train MAE	Test MAE	Train MSE	Test MSE	Train RMSE	Test RMSE	Train R2	Test R2
0	Linear Regression	885.5279	887.3977	1728808.4814	1773688.7744	1314.8416	1331.7991	0.8916	0.8877
1	Random Forest	451.8969	506.8259	659595.0975	862312.0541	812.1546	928.6076	0.9586	0.9454
2	XGBoost	473.3746	495.1529	737410.3019	804838.7079	858.7260	897.1280	0.9538	0.9491
3	CatBoost	473.5462	496.1716	716152.4999	807319.5544	846.2579	898.5096	0.9551	0.9489

From the figure, it can be seen that all 3 models vastly outperform the Linear Regression model. Looking at the data more closely, it can be seen that XGBoost slightly outperforms both CatBoost and Random Forest on the test set. Hence XGBoost was selected as the model of choice. The hyperparameters were then optimized using SciKit Learn's Randomized Search and the model was retrained with the best hyperparameters. The final price prediction model yielded the following metrics on the testing set, with the average shown for comparison purposes.

	Metric	Value
0	Mean Squared Error (MSE)	802920.3551
1	Mean Absolute Error (MAE)	494.8491
2	Root Mean Squared Error (RMSE)	896.0582
3	R^2 Score	0.9492
4	Average of y_train	3935.9530

This is slightly lower than the values obtained by [Diamond Price Prediction using Machine Learning], who obtained R^2 of 0.987, MAE of 271.10, and 525.81 with CatBoost. However, this loss of accuracy can be attributed to two reasons. The first is that this model does not use all of the columns present in the 'Diamonds' dataset, such as dimensions and table, since it is to be used with the neural network and thus only takes the 4 Cs as output. The second is that the encoding done within the neural network to reduce computational time, which involved mapping a range of values to a certain class had to be matched here, which results in the model losing some information regarding the diamond.


Residuals Plot
Residuals
Predicted Price
6000
4000
2000
0
−2000
−4000
−6000
−8000
0
2500
5000
7500
10000
12500
15000
17500


Actual vs Predicted Diamond Prices (Sampled)
Predicted Price
Actual Price
17500
15000
12500
10000
7500
5000
2500
0
0
2500
5000
7500
10000
12500
15000
17500

9. Discussion and Future Work

Our neural network model from our research marks the first attempt to predict the 4 C's of the diamond- Cut, Clarity, Color, and Carat from its image. While there have been some previous commercial attempts such as, they have only attempted to predict only one or two of the four C's of the diamond through different models like color classification of the diamond from its image using EfficientNet. Most of the research using machine learning or neural networks in the field of gemology and diamonds has been limited to either identifying the gem type or trying to derive the price of the diamond by using regression models that contain all the relevant information about the diamond without the need or use of an image. Hence being able to achieve relatively high accuracies with our final model was an immense success and showcased the potential for neural networks not just in the field of diamonds but overall gemology.

The EfficientNetB0 model was not very successful, since it did not obtain good accuracy metrics despite having undergone similar data augmentation. However it did not do well on the data, and we were constrained by time and computational resources so we could not test additional variants of EfficientNet. Future work taking that approach could use other configurations of EfficientNet. Additionally, improving the model accuracy could be done by applying transfer learning and taking the weights from our 2D CNN and applying it to EfficientNet to see if there is any increase in performance. The 2D CNN model was very successful and showcases the immense potential that CNNs can have in diamond grading.

However, we have barely scratched the well of potential that is the use of neural networks in gemology, let alone diamonds. Due to computational constraints, our training necessitated the mapping of multiple grades of diamonds to broader categories based on quality. Additionally, our dataset did not have enough samples for some grades of diamonds on the end of the spectrum. While we were able to fix some of it with image augmentation, we had to drop a few at the end of the spectrum and limit ourselves to the available data. Lastly, our study was limited to round diamonds as they formed the majority of the market and we wanted to focus on getting acceptable accuracy on a singular type of diamonds before moving to advanced categories. Even with these constraints, the model as seen in ROC curves in Fig 17 showed that the model was reliable enough to predict with relatively high accuracy.

Hence future work could continue by providing more computational power to the process, and thus being able to more specifically classify the 4 C's of diamonds. Additional layers can be added to neural networks and samples selected from all over the dataset so that it can also classify diamonds of different shapes. An alternative approach could be using all 4 models classifying all 4 C's individually from images, particularly for carat and cut since they have been neglected in such projects, and fine-tuning each model individually to produce a more accurate output. However, this would require vastly more computational power and training time.

10. Link to a YouTube Video

Predicting diamond 4 Cs from the image:

https://youtu.be/6mRJ4YdJE04

Price Prediction:

https://youtu.be/98j5cPE4zkI

References

[1] H. Fu, S. Wang, G. He, & et al., "A 3D convolutional neural network model with multiple outputs for simultaneously estimating the reactive transport parameters of sandstone from its CT images", *Artificial Intelligence in Geosciences*, pg. 100092, vol. 5, 2024, doi: https://doi.org/10.1016/j.aiig.2024.100092

[2] P. Khalilian, F. Rezaei, N. Darkhal, & et al. "Jewelry rock discrimination as interpretable data using laser-induced breakdown spectroscopy and a convolutional LSTM deep learning algorithm." *Sci Rep 14,* 2024, doi: https://doi.org/10.1038/s41598-024-55502-x

[3] N. Zhenxing, M. Zhou, W. Le, & et al., "Ordinal Regression with Multiple Output CNN for Age Estimation", *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pg. 4920-4928, 2016, doi: https://doi.org/10.1109/CVPR.2016.532

[4] T. Judixon, & R. Shivaraman, "Gemstone Classification using Convolutional Neural Network", *International Journal of Engineering Research & Technology*, vol. 12 Issue 05, May 2023, doi: http://dx.doi.org/10.1007/s40031-024-01003-4

[5] Z. Ren, J. Liao, & L. Cai, "Diamond color grading based on machine vision," *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pg. 1970-1976, 2009, doi: https://doi.org/10.1109/ICCVW.2009.5457523

[6] D. Swain, A. Chaturvedi, & V. Rathi, "GemInsight: Unleashing Random Forest for Diamond Quality Forecasting", *2023 7th International Conference on Computing, Communication, Control And Automation (ICCUBEA)*, pg. 1-4, 2023, doi: https://doi.org/10.1109/ICCUBEA58933.2023.10392209

[7] B. T. Hanh, H. V. Manh, & N. Nguyen, "Enhancing the performance of transferred efficientnet models in leaf image-based plant disease classification", *Jounral of Plant Diseases and Protection*, Vol. 129, pg. 623-634, 2022, doi: https://doi.org/10.1007/s41348-022-00601-y

[8] Yashu, V. Kukreja, K. Madan, & et al., "GemID: A Hybrid CNN-Random Forest Approach for Accurate Gemstone Identification", *2023 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)*, pg. 1-5, 2023, doi: https://doi.org/10.1109/SMARTGENCON60755.2023.10442920

[9] W. Wang, & L. Cai, "Inclusion extraction from diamond clarity images based on the analysis of diamond optical properties", *Opt. Express*, Vol. 27, pg. 27242-27255, Sep 2019, doi: https://doi.org/10.1364/OE.27.027242

[10] H. Izadi, J. Sadri, & M. Bayati, "An intelligent system for mineral identification in thin sections based on a cascade approach", *Computers & Geosciences*, Vol. 99, pg. 37-49, 2017, doi: https://doi.org/10.1016/j.cageo.2016.10.010

[11] H. P. Borges, & M. Aguiar, "Mineral Classification Using Machine Learning and Images of Microscopic Rock Thin Section", *Advances in Soft Computing*, pg. 63-76, Oct 2019, doi: https://doi.org/10.1007/978-3-030-33749-0_6

[12] W. M. Freire, A. M. M. M. Amaral, & Y. M. G. Costa, "Gemstone classification using ConvNet with transfer learning and fine-tuning", *2022 29th International Conference on Systems, Signals and Image Processing (IWSSIP)*, Vol. CFP2255E-ART, pg. 1-4, 2022, doi: https://doi.org/10.1109/IWSSIP55020.2022.9854441

[13] J. M. Alcalde-Llergo, E. Yeguas-Bolivar, A. Zingoni, et al., "Jewelry Recognition via Encoder-Decoder Models", *2023 IEEE International Conference on Metrology for eXtended Reality, Artificial Intelligence and Neural Engineering (MetroXRAINE)*, pg. 116-121, 2023, doi: https://doi.org/10.1109/MetroXRAINE58569.2023.10405609

[14] M. Hedyehzadeh., K. Maghooli, M. MomenGharibvand, et al., "A Comparison of the Efficiency of Using a Deep CNN Approach with Other Common Regression Methods for the

Prediction of EGFR Expression in Glioblastoma Patients.", *J Digit Imaging 33*, Dec 2020, doi: https://doi.org/10.1007/s10278-019-00290-4

[15] A. Murthy, Prathwini, & S. Kulkarni, "An Exploration of Deep Learning-Based Object Detection and Classification in Yakshagana Imagery with YOLOv3", *2024 5th International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pg. 991-997, 2024, doi: https://doi.org/10.1109/ICESC60852.2024.10689830

[16] D. Katkus, P. Maciulevičius, & A. Lipnickas, "Amber Gemstones Colour Classification by CNN", *2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, vol. 1, pg. 531-536, 2023, doi: https://doi.org/10.1109/IDAACS58523.2023.10348663

[17] H. Huang, & R. Cui, "Deep Multi-feature Fusion CNNs with Gemstone Image Classification Algorithm", *2022 14th International Conference on Signal Processing Systems (ICSPS)*, pg. 344-349, 2022, doi: https://doi.org/10.1109/ICSPS58776.2022.00066

[18] F. D. Feyter, B. Claes, & T. Goedemé, "Rotation Equivariance for Diamond Identification", *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 5: VISAPP*, pg. 115-123, 2023, doi: https://doi.org/10.5220/0011658400003417

[19] F. D. Feyter, D. Hulens, B. Claes, & et al., "Deep Diamond Re-ID", *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pg. 2020-2025, 2019, doi: https://doi.org/10.1109/ICMLA.2019.00323

[20] W. Yanjuan, "Sapphire Color Grading Method based on color Space Feature Automatic Clustering Algorithm", *2024 International Conference on Optimization Computing and Wireless Communication (ICOCWC)*, pg.1-4, 2024, doi: https://doi.org/10.1109/ICOCWC60930.2024.10470585

[21] J. Abedin, J. Antony, K. McGuinness, et al., "Predicting knee osteoarthritis severity: comparative modeling based on patient's data and plain X-ray images", *Scientific Reports*, vol. 9, Apr 2019, doi: https://doi.org/10.1038/s41598-019-42215-9

[22] X. Zhao, L. Wang, Y. Zhang, et al., "A review of convolutional neural networks in computer vision", *Artifical Intelligence Review*, Vol. 57, 2024, doi: https://doi.org/10.1007/s10462-024-10721-6

[23] K. Fujiwara, "Anomaly detection for synthetic diamond grain using monocular depth estimation", *Computers & Industrial Engineering*, Vol. 188, pg. 109904, doi: https://doi.org/10.1016/j.cie.2024.109904

[24] M. Aljazaeri, Y. Bazi, H. AlMubarak, & et al., "Faster R-CNN and DenseNet Regression for Glaucoma Detection in Retinal Fundus Images", *2020 2nd International Conference on Computer and Information Sciences (ICCIS)*, pg. 1-4, 2020, doi: https://doi.org/10.1109/ICCIS49240.2020.9257680

[25] K. Zhang, W. Wang, Z. Lv, & et al., "A CNN-based regression framework for estimating coal ash content on microscopic images", *Measurement*, Vol. 189, pg. 110589, 2022, doi: https://doi.org/10.1016/j.measurement.2021.110589

[26] C. F. Sabottke, M. A. Breaux, & B. M. Spieler, "Estimation of age in unidentified patients via chest radiography using convolutional neural network regression", *Emergency Radiology*, Vol. 27, pg. 463-468, Apr 2020, doi: https://doi.org/10.1007/s10140-020-01782-5

[27] Z. OuYang, "Research on the Diamond Price Prediction based on Linear Regression, Decision Tree and Random Forest", *Highlights in Business, Economics and Management*, Vol. 24, pg. 248-257, 2024, doi: https://doi.org/10.54097/13ccwv59

[28] M. S. Sánchez, "Optimization of diamond price prediction strategies using machine learning techniques", *Journal of Economics and International Finance*, Vol. 16(3), pg. 28-38, Jul 2024, doi: https://dx.doi.org/10.5897/JEIF2024.1223

[29] S. Chu, "Pricing the C's of Diamond Stones", *Journal of Statistics Education*, Vol. 9, 2001, doi: http://dx.doi.org/10.1080/10691898.2001.11910659

[30] A. C. Pandey, S. Misra, & M. Saxena, "Gold and Diamond Price Prediction Using Enhanced Ensemble Learning", *2019 Twelfth International Conference on Contemporary Computing (IC3)*, pg. 1-4, 2019, doi: http://dx.doi.org/10.1109/IC3.2019.8844910

[31] H. Mihir, M. I. Patel, S. Jani, & et al., "Diamond Price Prediction using Machine Learning", *2021 2nd International Conference on Communication, Computing and Industry 4.0 (C2I4)*, pg. 1-5, 2021, doi: http://dx.doi.org/10.1109/C2I454156.2021.9689412

[32] S. N. Njoroge, E. Otieno, & B. Omolo, "Assessing predictive performance of supervised machine learning algorithms for a diamond pricing model", *Scientific Reports 13*, 2023, doi: http://dx.doi.org/10.1038/s41598-023-44326-w

[33] A. Farhoumand, “GIA Diamond Grading Scales: The Universal Measure of Quality,” *GIA 4Cs*, Mar. 14, 2013. https://4cs.gia.edu/en-us/blog/gia-diamond-grading-scales/

[34] “Ringo: World’s First AI Diamond Selection Tool,” *The Diamond Pro*, Dec. 21, 2023. https://www.diamonds.pro/ringo/ (accessed Dec. 05, 2024).