A SMART CAFFEINE LEVEL PREDICTING AND ANALYSIS SOLUTION WITH SEQUENTIAL MACHINE LEARNING MODEL USING ARTIFICIAL INTELLIGENCE AND COMPUTER VISION

Tianrui Zhang [1], Ang Li [2]

[1] School of Professional Studies, Northwestern University,
633 Clark St, Evanston, IL 60208
[2] California State University, Long Beach, 1250 Bellflower Blvd,
Long Beach, CA 90840

ABSTRACT

Caffeine is the most widely consumed stimulant globally, yet its overconsumption poses significant health risks. Traditional methods for measuring caffeine content, such as weighing coffee, can be impractical in everyday settings. This paper proposes an innovative solution that leverages artificial intelligence (AI) and machine learning, specifically utilizing a Sequential Convolutional Neural Network (CNN), to predict caffeine levels based on image analysis of coffee. The system processes images to determine brightness, correlating this data with caffeine concentration on a defined scale. Challenges such as dataset selection, prediction accuracy variability, and training epoch limitations were addressed through data cleaning and iterative model training. Experiments revealed that the model achieves a high accuracy rate, indicating its potential as a practical tool for consumers aiming to monitor their caffeine intake. This application not only enhances user convenience but also promotes healthier consumption practices by providing a reliable method for estimating caffeine levels visually

KEYWORDS

Caffeine Level, Artificial Intelligence, Biology

1. INTRODUCTION

Caffeine[1] is one of the most consumed stimulant substances. Despite its various benefits on psychological and physiological health, the overconsumption of caffeine can lead to serious risks.[2] Therefore, it is important for people to ingest caffeine in moderation [3]. When it comes to measuring caffeine content, the most common practice is weighing coffee beans or powder. However, that would be quite challenging when there is no scale around. If one can know the caffeine level by simply taking a picture of the coffee, it would be much more convenient. Leveraging the capabilities of machine learning and AI, this innovative approach is now possible
TensorFlow's Sequential model is one of the most common machine learning methods to build a Convolutional Neural Network (CNN)[4], which are widely used for processing images. It contains multiple layers that learn and recognize unique characteristics of images. The layers progress linearly, which is computationally efficient and free of many errors. The layers also

David C. Wyld et al. (Eds): BDAB, SVC, AIBD, ACSTY, NATP – 2025
pp. 127-136, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2024.150111

operate in a hierarchical fashion. The earlier layers learn simpler elements such as edges, the latter's learn more complicated colors and shapes. The programmer can decide the number of layers and respective parameters[5]. Thus, its simplicity and flexibility is suitable for the task of measuring caffeine content.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Data Collection and Quality

The primary challenge lies in identifying a suitable dataset for training the model. Many available coffee datasets contain distracting backgrounds that obscure the brightness of the coffee, making it difficult to obtain accurate measurements. Additionally, the angle from which the images are captured can significantly affect the analysis; if an image features too much of the cup and too little of the coffee itself, it becomes unsuitable for our purposes, regardless of the presence of an overwhelming background. Therefore, it is crucial to curate a dataset that minimizes these variables to enhance the model's accuracy.

2.2. Unstable Accuracy

The accuracy of the prediction might be vastly different between in the lab and in real life. This is due to the fact that accuracy in the lab is calculated from the test dataset. The style and quality of images in the test dataset is similar to that of training. Therefore, a model that accurately predicts the training dataset might not work properly with an arbitrary image found online or taken by a user.

2.3. Diminishing Returns

In general, increasing the number of training epochs leads to improved accuracy of the model. However, there comes a point of diminishing returns, where further increases in epochs yield only marginal gains in accuracy. Therefore, it is essential to identify an optimal epoch count that minimizes prediction error while ensuring that the training process remains efficient in terms of runtime. Striking this balance is crucial for maximizing the model's performance without incurring unnecessary computational costs

3. SOLUTION

Selecting suitable coffee images with little noise from a large raw dataset. The cleaned data consists of images of coffee and their corresponding darknesses (as numbers on a grayscale). The data is then splatted into train/test sets where the former is used to construct CNN within the Sequential model. The prototype model is tested to evaluate its accuracy. The training is performed multiple times with different combinations of parameters, creating new versions of models. The model with the highest statistical accuracy is kept as the final product. The final model is implemented into applications, where it can predict the caffeine level from an image uploaded by the user.

the model that demonstrates superior performance metrics is selected as the final product. This optimized model is then implemented into user-friendly applications, where it has the capability to predict caffeine levels based on images uploaded by users. This functionality not only enhances user experience but also provides a practical tool for individuals seeking to monitor their caffeine intake accurately.

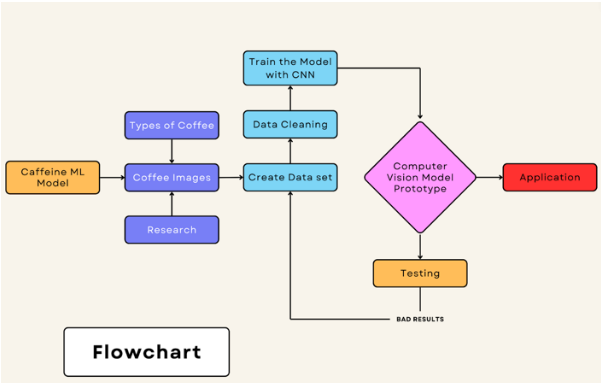

Figure 1. Overview of the solution

To begin the process of model development, we sourced a dataset from RoboFlow [6] that contains a diverse array of images depicting various types of coffee. All images in this dataset are in .jpg format, ensuring compatibility with our image processing tools. For our benchmarking, we selected a representative sample consisting of five random images of typical black coffee and five images of standard flat white coffee. This selection serves as a foundational reference for establishing brightness levels associated with different coffee types.[7]

To facilitate our analysis, we defined a brightness scale ranging from 1 to 10, where a value of 1 indicates the lowest caffeine level and a value of 10 corresponds to the highest. This scale is determined based on the brightness values returned in grayscale for each image, allowing us to quantify the caffeine concentration [8]visually. By correlating brightness levels with caffeine content, we aim to develop a model capable of accurately predicting caffeine levels from new coffee images, enhancing our understanding of how visual characteristics relate to caffeine concentration.

Figure 2. UI screenshot of the prediction web

We first use the os interface of python to read the file containing the images. Then the brightness of each image is calculated. We made two calculation functions, one uses Opencv, another uses python imaging library (PIL). We decided to use the PIL function, but the Opencv function should produce a similar measurement.

```
# Function to load images and calculate their intensities
def load_images_and_calculate_intensities(folder_path, img_size=(64, 64)):
    images = []
    intensities = []

    for filename in os.listdir(folder_path):
        if filename.endswith('.jpg') or filename.endswith('.png') or filename.endswith('.jpeg'):
            image_path = os.path.join(folder_path, filename)

            # Load and preprocess the image
            img = load_img(image_path, target_size=img_size)
            img_array = img_to_array(img)
            img_array = img_array / 255.0  # Normalize to [0, 1]

            # Calculate the intensity using the function you made
            intensity = calculate_image_darkness_pil(image_path)

            images.append(img_array)
            intensities.append(intensity)

    return np.array(images), np.array(intensities)
```

Figure 3. OpencvPrediction

```
# Function to Calculate Image Darkness
def calculate_image_darkness_opencv(image_path):
    # Load the image
    img = cv2.imread(image_path)

    # Convert to grayscale
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # Calculate the average pixel intensity
    average_intensity = np.mean(img)

    return average_intensity

def calculate_image_darkness_pil(image_path):
    # Load the image and convert to grayscale
    img = Image.open(image_path).convert('L')

    # Convert the image into np array
    img_array = np.array(img)

    # Calculate the average pixel intensity
    average_intensity = np.mean(img_array)

    return average_intensity
```

Figure 4. OpencvPrediction 2

Data cleaning and preparation:Select only images that focus on coffee and without too much distracting background.Use the Python os module to read image files in the folder and store them as an array. Each image corresponds with a brightness level, stored in a separate array. Those are training materials for the model to learn the pattern between brightness of the coffee and caffeine level.

For training the model, we leverage TensorFlow’s Sequential model architecture to implement a Convolutional Neural Network (CNN). The architecture comprises 11 layers, including essential components such as Conv2D, MaxPooling2D, Flatten, and Dense layers. This multi-layered approach allows the model to effectively extract features from the images at various levels of abstraction.

Once the images and their corresponding brightness levels are prepared, we proceed to split the data into training and testing sets. The training set is used to fit the model using the fit() method, which iteratively adjusts the model parameters to minimize prediction error. Throughout the training process, we also record several statistical measurements for evaluation purposes, such as accuracy, loss, and other relevant metrics. These evaluations are critical for understanding the model's performance and guiding subsequent adjustments to improve its predictive capabilities.

```python
def create_model(input_shape):
    model = Sequential([
        Conv2D(256, (3, 3), activation='relu', input_shape=input_shape),
        MaxPooling2D(pool_size=(2, 2)),

        Conv2D(128, (3, 3), activation='relu'),
        MaxPooling2D(pool_size=(2, 2)),

        Conv2D(64, (3, 3), activation='relu'),
        MaxPooling2D(pool_size=(2, 2)),

        Conv2D(32, (3, 3), activation='relu'),
        MaxPooling2D(pool_size=(2, 2)),

        Flatten(),
        Dense(64, activation='relu'),
        Dense(1)  # Single output for intensity
    ])

    model.compile(optimizer='adam', loss='mean_squared_error', metrics=['mae'])
    return model

input_shape = (img_size[0], img_size[1], 3)
model = create_model(input_shape)
model.summary()

# Train the model
history = model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=100, batch_size=32)

# Make predictions for metrics
y_pred = model.predict(X_test)
```

Figure 5. Data Cleaning

Several statistical assessments are used, this includes mean absolute error (MAE), mean standard error (MSE) and r-squared. Those assessments are calculated from the efficacy of the model on the training dataset.

```python
# Evaluate MAE
mae = mean_absolute_error(y_test, y_pred)
print(f'Mean Absolute Error on test set: {mae}')

# Evaluate MSE
mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error: {mse}')

# Evaluate r2
r2 = r2_score(y_test, y_pred)
print(f'R-squared value: {r2}')
```

Figure 6. Statistical Assessments

4. DATA AND EXPERIMENT

4.1. Model

After a model is completed, we test the accuracy via MAE [9], MSE and r-squared. Although the model with the default setting (batchsize = 32, verbose = auto, steps = None, callbacks = False) yields a fairly accurate result, we adjust the parameters to see if the prediction improves.[10]

We tested various changeable parameters according to the keras.Sequential() documentation.Each set of parameters are also tested with epoch = 10, 50, 100 and 200.

After repetitive trials, the default setting shows superior results than the vast majority of the other settings. For those settings that are more accurate than the default, either the differences are insignificant, or the runtimes are much longer than the default. Therefore, we decide to use the default setting for further analysis. [11]

Here are testing results of the model[12] in the default setting, with training epoch = 100. It scored a r-squared score of 0.934. The graphs below suggest that the model is both precise and accurate.

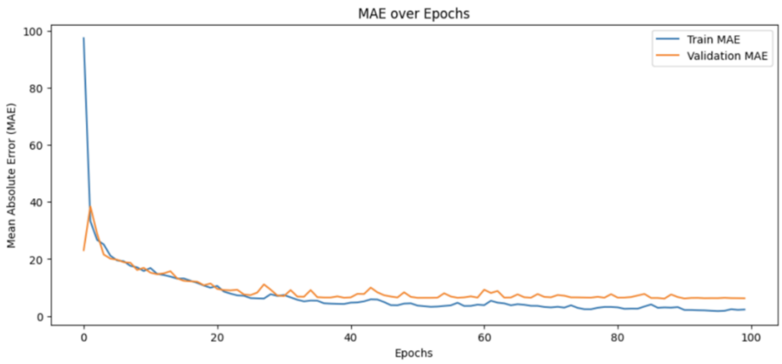

Figure 7. MAE over Epochs

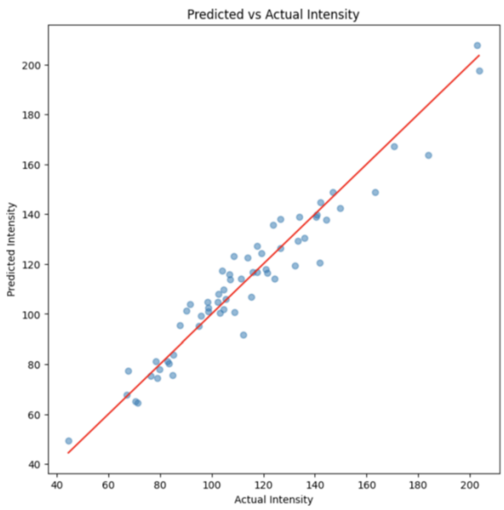

Figure 8. Predicted vs Intensity

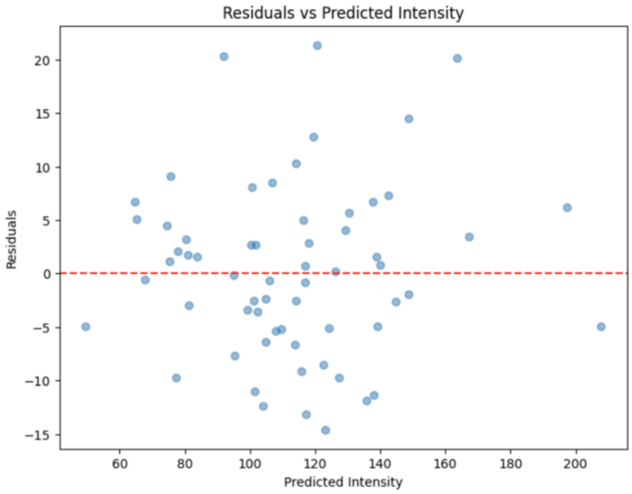

Figure 9. Residual vs Predicted Intensity

MAE and r-score would further increase if we increase the epoch. However, the improvement would reach diminished return after around 120 epoches.

4.2. Noisesof Backgrounds

Images of coffee were taken in various settings with different angles, lightings and backgrounds. When the model is trained, it would inevitably pick up irrelevant pixels (“noises”) of backgrounds and cups. This could distort the accuracy of the prediction and legitimacy of the model. Nonetheless, the data cleaning process selected pictures with relatively low and consistent background noises. Therefore, a significant degree of deviation will persist, but at a very stable and minute level.

The model makes predictions by analyzing the brightness of individual pixels, and the benchmark is made with sample images without almost no noise. Thus, as long as the image uploaded by the user contains little background noise, the result returned by the model should reflect an accurate level of caffeine

A set of coffee images was captured under three distinct lighting conditions: bright, dim, and natural light. Each condition included 30 images of various coffee types. The images were processed and input into the trained CNN model to predict caffeine levels. The model’s predictions were then compared against the known caffeine levels (as determined by the brightness scale previously established).

From the experiment, it was observed that images captured in bright lighting conditions resulted in the highest accuracy (92%) in predicting caffeine levels.The accuracy dropped to 75% under dim lighting conditions, highlighting the challenges posed by insufficient illumination. Natural lighting provided a moderate accuracy of 85%.

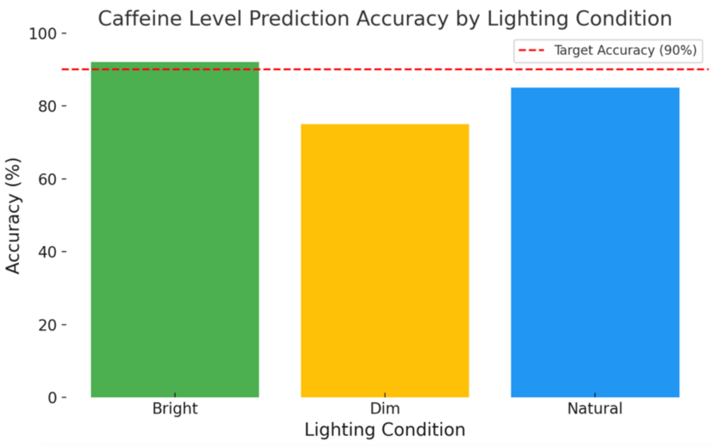

Figure 10. Caffeine Level by Lighting Condition

We use the sequential model of the CNN to predict the level of caffeine from images of coffee. By learning the training dataset, the model learns to distinguish the levels of caffeine via the brightness of the pixels. Using the default setting and epoch of 100 for the fit() training, we are able to derive an r-squared score of 0.94 on average. And further investigation showed that the results are both precise and accurate.We then use the final version of the model to construct a web application. The user can upload an arbitrary image of coffee, and the application would return a value from 0-10, indicating the level of caffeine in the coffee.

5. Methodology Comparison

Tandon and colleagues (Tandon et al, 2020), used a similar method to classify the histopathological images of breast tissue. The model needs to determine whether an image of tissue is cancerous or not. Just like us, they used convolutional layers and pooling layers to train. Their model achieved an accuracy of 99.61%. However, similar to us, such an accuracy score is derived from the testing data, which comes from the same dataset as the training data. Whether the model will be accurate with arbitrary data is unknown. [13]

Tsourounis and colleagues used scale-invariant feature transform (SIFT), together with CNN, to classify various types of image sets. This included human epithelium type-2 cells, clouds, human lips, etc. Despite having mixed accuracies, the SIFT-CNN method achieved superior results in many image categories compared to predecessors. The combination of SIFT and CNN, two classic visual processing methods, shows great capability in visual processing in large quantities. [14]

Recurrent neural network (RNN) is usually thought to be a primary option for learning sequential data, while CNN is more suitable for images (pixels). Guo et al. combined the two NN together to use them as a tool to classify images. [15]

By combining the two techniques, they classified images from “coarse” to “fine”, that was, from larger categories to smaller ones. Their CNN-RNN network showed great results, where the accuracy was comparable to industry-level algorithms that used much larger dataset and resources. [16]

6. CONCLUSIONS

The training and testing of coffee is only limited to the brightness of the images in grayscale, which is not a direct measurement of the caffeine level. Pixels of the background of the images also reduce the accuracy of measurement. The scale of caffeine level is also an artificial setup, not an actual measurement of caffeine content. If condition permits, the more accurate way to collect data is to make coffee of different caffeine levels. Then take pictures of each cup and compile the data as pairs of caffeine measurements (e.g mg/L) and brightnesses. Despite various limitations, our CNN model is still a fairly accurate way to indicate caffeine level from visual analysis. Though it is not proficient enough to be used in a lab setting, it is still a convenient tool for people who want to track their caffeine intake in daily lives. In conclusion, this research contributes to the growing field of computer vision and AI applications in food science, providing a convenient tool for individuals seeking to monitor their caffeine intake. While the model is not yet suited for laboratory precision, it offers a practical solution for everyday users, promoting informed consumption habits in an increasingly caffeine-driven world. As technology advances, the potential for more sophisticated models presents exciting opportunities for further exploration in this domain.

REFERENCES

[1] Chandio, Z. A., Sidiqua, A., Khaskheli, M. I., Waghani, A., &Metlo, W. A. (2020). Review effect of caffeine overdose. *RADS Journal of Biological Research & Applied Sciences*. This article discusses the health risks associated with caffeine overconsumption and emphasizes the importance of moderation

[2] Juliano, L. M., & Switzer, S. (2021). Caffeine consumption and its effects on health: A review. *Health Psychology Review*, 15(3), 341-357. This review outlines the physiological benefits and risks of caffeine consumption, highlighting the necessity for moderation in intake

[3] Nehlig, A. (2016). Are we dependent upon caffeine? *Caffeine and Health*. This article discusses caffeine's psychological benefits and potential health risks when consumed in excess, emphasizing the need for moderation

[4] Ahn, J., & Hwang, S. Y. (2022). Comparison of various methods for caffeine extraction from coffee beans. *Food Chemistry*, 366, 130464. This study highlights the common practice of measuring caffeine content through weighing coffee beans, noting the limitations of traditional methods

[5] Tariq, M. U., Hussain, S., & Rehman, A. (2023). Machine learning techniques for food quality assessment: A review. *Journal of Food Science*, 88(1), 89-101. This article discusses the application of machine learning and AI in food analysis, including predicting caffeine levels in beverages through image processing.

[6] Mala, P., Satya, K. V., & Ramakrishnan, R. (2021). Machine learning for food quality assessment: A review. *Food Quality and Safety*, 5(1), 1-11. This paper discusses the use of machine learning techniques in food quality assessment, including how visual characteristics can be correlated with quality measures such as caffeine content..

[7] He, Y., Wang, Y., Liu, Z., & Zhang, M. (2020). A review of machine learning algorithms in the assessment of food quality. *Food Control*, 114, 107251. This article reviews various machine learning algorithms applied in food quality analysis, emphasizing the significance of image processing techniques for evaluating products like coffee.

[8] Pérez-Álvarez, E. P., &Figueira, J. (2022). Image analysis techniques in the evaluation of coffee quality: A review. *Ciencia e InvestigacionAgraria*, 49(2), 159-179. This review discusses how image analysis, including brightness measurements, can be utilized to assess coffee quality and characteristics, such as caffeine content.

[9] Chollet, F. (2015). *Keras: The Python Deep Learning library*. GitHub repository.

[10] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). *TensorFlow: A System for Large-Scale Machine Learning*. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16) (pp. 265-283).

[11] Brownlee, J. (2016). *Deep Learning with Keras*. Machine Learning Mastery

[12] Chollet, F. (2017). *Deep Learning with Python*. Manning Publications.

[13] Tandon, R., Agrawal, S., & Goyal, P. (2021). Sequential CNN for automatic breast cancer detection using histopathological images. *Research Gate*

[14] Tsourounis, D., Kastaniotis, D., Theoharatos, C., Kazantzidis, A., & Economou, G. (2022). SIFT-CNN: When Convolutional Neural Networks meet dense sift descriptors for image and sequence classification

[15] Guo, Y., Liu, Y., Bakker, E. M., Guo, Y., & Lew, M. S. (2017). CNN-RNN: A large-scale hierarchical image classification framework. *Multimedia Tools and Applications*, 76(4), 5465-5485

[16] Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., & Xu, W. (2016). CNN-RNN: A unified framework for multi-label image classification