February 22, 2024

C964: Computer Science Capstone Template

Task 2 parts A, B, C and D

[Part A: Letter of Transmittal 2](#_Toc159513543)

[Part B: Project Proposal Plan 4](#_Toc159513544)

[Project Summary 4](#_Toc159513545)

[Data Summary 5](#_Toc159513546)

[Implementation 6](#_Toc159513547)

[Timeline 7](#_Toc159513548)

[Evaluation Plan 7](#_Toc159513549)

[Resources and Costs 9](#_Toc159513550)

[Part C: Application 9](#_Toc159513551)

[Part D: Post-implementation Report 9](#_Toc159513552)

[Solution Summary 9](#_Toc159513553)

[Data Summary 10](#_Toc159513554)

[Machine Learning 11](#_Toc159513555)

[Validation 13](#_Toc159513556)

[Visualizations 14](#_Toc159513557)

[User Guide 14](#_Toc159513558)

[Reference Page 15](#_Toc159513559)

# Part A: Letter of Transmittal

February 22, 2024

Ronald MacDonald

Translatinator, LLC

123 Sunshine Road Los Angeles CA, 90001

Dear Ronald MacDonald

I hope this message finds you well. I am writing to propose an exciting enhancement to our language-learning application, Translatinator, which I believe will significantly distinguish our product from competitors such as RosettaStone and DuoLingo. Despite offering a high-quality product at an excellent price point, Translatinator currently lacks unique features that set it apart in the crowded language-learning space.

**Proposal Overview:**

I propose the integration of a cutting-edge feature that leverages machine learning technology. This feature will enable users to take pictures of common objects, specifically cats and dogs in the initial phase, and receive the name of the object in the language they are learning. This approach not only adds a novel functionality to our app but also aligns with our mission to provide intuitive and innovative language-learning solutions.

**Benefits to Translatinator:**

***Enhanced Functionality***: This feature will enrich the user experience by offering a practical, real-world application of language learning, thus increasing user engagement and satisfaction.

***Technological Relevance***: By incorporating machine learning, we underscore our commitment to innovation, keeping Translatinator at the forefront of technology trends in educational applications.

***Intuitive Learning***: This enhancement aligns with our goal to create a natural and immersive learning environment, making the acquisition of new vocabulary more intuitive and engaging.

**Project Summary:**

***Costs***: The project entails 200 hours of developer time at $40/hour and 100 hours of project management at $30/hour, with additional costs for server usage at $500/month to train and store the AI model. The software framework, TensorFlow, is available at no cost.

***Timeline***: We aim to complete this project within 6 weeks, ensuring a swift enhancement to our app's capabilities.

***Data***: The development will utilize 3,500 labeled images of cats and dogs, providing a robust dataset for initial training.

**My Expertise:**

As a computer science student with extensive experience in machine learning applications, including proficiency in Python, AIML, and application design, I am well-equipped to lead this project to success. My background ensures a deep understanding of the technological and practical aspects necessary for this enhancement.

I am confident that this feature will not only improve our competitive edge but also demonstrate our ongoing commitment to innovation and customer value. I look forward to discussing this proposal further and am available for any questions or additional information you may require.

Thank you for considering this proposal. I am excited about the potential impact this project could have on Translatinator’s market position and our users' learning experiences.

Sincerely,

Andrew Steurer

Andrew Steurer, Intern

# Part B: Project Proposal Plan

## Project Summary

* Summary of the Problem
  + Translatinator, a competitive language-learning application in the market alongside giants like RosettaStone and DuoLingo, presents a high-quality product at an exceptional price. Despite these advantages, Translatinator struggles to stand out due to its lack of distinctive features. This challenge hampers its ability to attract new users and retain existing ones, necessitating a solution that enhances its appeal and functionality. Implementing a feature that will allow users to take pictures of common objects and receive the translation in the language they are learning will be a powerful solution to this problem.
* Company Needs
  + Translatinator requires a novel approach to elevate their market position and differentiate it from the competition. The core need is to integrate a feature that not only adds value to the user experience but also leverages current technological trends to enhance learning outcomes.
* Deliverables
  + Machine Learning Model: A robust model will be trained to recognize images of cats and dogs with high accuracy.
  + Application: A command line interface that will allow the user to interact with the machine learning model.
  + User Guide: A set of instructions for how to set up the environment and how to run the application.
* Summary of Benefits
  + Added Functionality: By enabling users to learn words through object recognition, Translatinator will add a unique, interactive element to language learning that is currently absent in competitor apps.
  + Technological Relevance: Incorporating machine learning for object recognition signals to current and potential users that Translatinator is at the forefront of applying emerging technologies to enhance learning experiences.
  + Intuitive Learning Experience: This feature will offer a more natural, engaging way for users to learn new vocabulary, mimicking the way languages are learned naturally through interaction with the environment.

## Data Summary

* The categorized image data will be sourced from two data repositories hosted on kaggle.com:
  + <https://www.kaggle.com/datasets/alifrahman/dataset-for-wbc-classification>
  + <https://www.kaggle.com/datasets/samuelcortinhas/cats-and-dogs-image-classification>
* Development Life Cycle Phases:
  + Design:
    - Data Requirement Analysis: Identify the types of data needed for the machine learning model, specifically focusing on images of cats and dogs. This includes understanding the diversity in breeds, sizes, and environments to ensure the model is well-trained.
    - Data Privacy and Security Assessment: Evaluate the implications of handling user-uploaded images, ensuring compliance with privacy regulations. This involves designing data anonymization processes and secure data storage solutions.
    - Data Collection Plan: Outline methods for collecting training data, ensuring it's ethically sourced and respects copyright laws.
  + Development:
    - Data Preparation and Preprocessing: Clean, annotate, and prepare the collected images for training. This process includes categorizing images, removing identifiable information, and possibly augmenting the dataset to improve model robustness.
    - Model Training and Testing: Utilize the processed data to train the machine learning model. This involves iterative testing and validation against a test dataset to ensure accuracy and minimize bias.
    - Application Integration: Build a command line interface that interacts with the model and allows testers to provider their own images to test.
  + Maintenance:
    - Continuous Data Monitoring: Implement monitoring tools to track the performance of the object recognition feature and identify any issues with data processing or model accuracy.
    - Data Storage Management: Regularly review data storage practices to ensure they remain secure and compliant with evolving data protection laws. This includes periodic audits and updates to data encryption and anonymization techniques.
    - Model Updates and Re-training: As new data becomes available or user feedback highlights areas for improvement, re-train the model with new images or refined algorithms. This may include expanding the model's capabilities to recognize additional objects beyond cats and dogs.
* Justify why the data meets the needs of the project.
  + The core functionality of the new feature is to recognize images of common objects (initially, cats and dogs) and translate their names into the language a user is learning. Thus, a dataset comprising diverse images of cats and dogs is directly relevant as it serves as the foundation for training the machine learning model to accurately identify these objects. Focusing on cats and dogs makes the project more manageable and achievable within its scope. It ensures that the data collection, processing, and model training phases are practical and feasible, providing a clear pathway to successful implementation and deployment. This focus allows for a concentrated effort on achieving high accuracy and user satisfaction with the feature before expanding to more complex objects or categories.
* Address any ethical or legal concerns regarding the data.
  + There are no ethical concerns with the current data. The data is sourced from a free, public data repository that contains images of dogs and cats only.

## Implementation

* Describe an industry-standard methodology to be used.
  + For the Translatinator enhancement project, implementing the feature that allows users to take pictures of common objects (initially cats and dogs) and receive the translation in the language they are learning, an industry-standard methodology that would be particularly effective is Agile development with Scrum. This methodology is well-suited for projects requiring flexibility, rapid iteration, and close collaboration between developers, project managers, and stakeholders. Agile development emphasizes iterative progress, flexibility, and the delivery of functional product components in short cycles called sprints. Scrum, a subset of Agile, provides a structured framework for managing this process, focusing on quick deliveries, and incorporating feedback to refine and adjust the project direction as needed.
* An outline of the project’s implementation plan.
  + Phase 1: Pre-Development
    - Define the scope and specific objectives for the machine learning solution.
    - Identify the languages and the number of objects (initially cats and dogs) to be recognized.
  + Phase 2: Development
    - Set up the development environment and tools.
    - Develop initial prototypes of the machine learning model using a subset of the data.
    - Evaluate prototype models to select the most promising approach.
    - Expand model training using the full dataset.
    - Implement model validation and tuning to improve accuracy and reduce overfitting.
    - Build a CLI application that interacts with the model.
  + Phase 3: Testing
    - Conduct comprehensive testing including unit tests, integration tests, and user acceptance testing.
    - Collect feedback from testers to identify issues and areas for improvement.
  + Phase 4: Deployment and Monitoring
    - Finalize documentation.
    - Deploy the enhanced feature to the production environment.
    - Monitor the feature's performance and user feedback closely.
    - Address any immediate issues or bugs reported by users.
  + Phase 5: Improvements
    - Continue to iterate on the machine learning model and feature based on ongoing user feedback and technological advancements.
    - Plan for the expansion of the object recognition capability to include more objects beyond cats and dogs, based on user demand and success metrics.

## Timeline

|  |  |  |  |
| --- | --- | --- | --- |
| Milestone or deliverable | Duration  (hours or days) | Projected start date | Anticipated end date |
| Trained ML Model | 2 days | 02/24/2024 | 02/26/2024 |
| CLI Application | 1 day | 02/27/2024 | 02/28/2024 |
| User Guide | 2 hours | 02/29/2024 | 02/29/2024 |

## Evaluation Plan

* Describe the verification method(s) to be used at each stage of development.
  + Phase 1: Pre-Development
    - Requirements Validation: Confirm that the project objectives, scope, and machine learning solution requirements accurately reflect stakeholder needs and expectations.
    - Data Readiness Assessment: Ensure the collected dataset is of high quality, diverse, and sufficient for training the machine learning model. This includes checking for biases, missing values, and data diversity.
  + Phase 2: Development
    - Model Prototyping: Evaluate the feasibility of different machine learning algorithms and select the most promising approach based on performance metrics such as accuracy and processing time.
    - Model Development and Training: Verify the model's accuracy, generalization ability, and robustness across different subsets of the data to prevent overfitting and underfitting.
    - Integration With CLI Application: Ensure the machine learning model's API interfaces correctly with the Translatinator app.
  + Phase 3: Testing
    - Testing: Validate individual components (unit testing), their interactions (integration testing), and the overall solution (UAT) to ensure it meets user expectations and requirements.
    - Assess and enhance the machine learning model and app feature's performance, based on quantitative benchmarks and qualitative user feedback.
  + Phase 4: Deployment and Monitoring
    - Preparation: Ensure all elements of the feature are ready for launch, including final testing of the deployment environment and infrastructure.
    - Launch: Immediately identify and address any issues upon release, monitoring system performance and user engagement in real time.
    - Post-Launch Support: Ongoing evaluation of the feature's performance and user satisfaction to inform future improvements and adjustments.
  + Phase 5: Improvements
    - A/B Testing: Test new updates or features against the current version to evaluate improvements or changes in user engagement and satisfaction, guiding future development priorities.
* Describe the validation method to be used upon completion of the project.
  + Technical Accuracy Validation
    - Model Accuracy Metrics: Utilize standard machine learning metrics such as precision, recall, F1 score, and accuracy to evaluate the model's performance on a validation dataset. These metrics will provide insights into how well the model identifies cats and dogs, balancing between correctly labeling the objects and minimizing false positives or negatives.
    - Confusion Matrix Analysis: Use a confusion matrix to visualize the model's predictions versus the actual labels, identifying any patterns of misclassification. This analysis is particularly useful for determining if the model struggles with specific breeds or lighting conditions.
  + User Experience Validation
    - User Acceptance Testing (UAT): Conduct UAT with a diverse group of users to assess the feature's usability, intuitiveness, and overall satisfaction. Participants will be asked to use the object recognition feature under various conditions and provide feedback on their experience.
    - Beta Testing: Release the feature to a limited user base to gather real-world usage data and feedback. Beta testing will help identify any unforeseen issues or usability challenges not captured during UAT.
    - Surveys and Feedback Forms: Distribute surveys and feedback forms to both UAT participants and beta testers to collect quantitative and qualitative feedback on the feature's performance, user interface design, and overall impact on the language learning experience.
  + Combined Validation Approaches
    - A/B Testing: If feasible, conduct A/B testing with the new feature, presenting it to one group of users while withholding it from another. This method can validate the feature's effectiveness in enhancing the language learning experience by comparing engagement metrics, retention rates, and learning outcomes between the two groups.
    - Analytics and Usage Metrics: Analyze usage metrics such as frequency of feature use, average session length, and progression in learning objectives when the feature is used. These metrics will validate the feature's impact on user engagement and learning efficiency.
    - Continuous Monitoring and Feedback Loop: After the feature's public release, continue to monitor user feedback and performance metrics. This ongoing validation process ensures that any issues are promptly addressed and that the feature continues to meet user needs and expectations.

## Resources and Costs

* Developer Costs
  + Developer time: 200 hours at $40/hour = $8,000
* Project Management Costs
  + Project Manager: 100 hours at $30/hour = $3,000
* Hardware Costs
  + Servers to train and store the model: $500/month
* Software Costs
  + Software (TensorFlow): Free
* Total Project Costs:
  + Developer Costs: $8,000
  + Project Management Costs: $3,000
  + Hardware Costs: $500/month (for the duration of the project and ongoing maintenance)
  + Software Costs: $0 (TensorFlow is free)

**Part C: Application**

# Part D: Post-implementation Report

## Solution Summary

* Summarize the problem and solution.
  + Translatinator faced a challenge in distinguishing itself in the crowded language-learning app market due to its lack of unique features, which affected its ability to attract and retain users despite offering a high-quality product at an attractive price. To address this issue, the proposed solution involved integrating a new feature that allowed users to take pictures of common objects, initially focusing on cats and dogs, and used machine learning to identify these objects and provided their names in the language the user is learning. This innovative approach not only added significant value to the user experience by leveraging current technological trends but also positioned Translatinator as a forward-thinking application, enhancing its appeal and distinguishing it from competitors.
* Describe how the application provided a solution to the problem from parts A and B.
  + The application provided a solution to Translatinator's problem of lacking distinctive features in the competitive language-learning market by introducing an innovative object recognition feature powered by machine learning. This feature allowed users to take pictures of common objects, starting with cats and dogs, and the application then identified these objects and provided their names in the language the user is learning. This integration accomplished several key objectives:
    - Enhanced User Engagement: By enabling an interactive way of learning where users can connect language learning with their immediate environment, the feature made the learning process more engaging and practical. This hands-on approach is likely to increase user retention and attract new users looking for a more dynamic and interactive learning experience.
    - Leveraged Technological Trends: Utilizing machine learning for object recognition places Translatinator at the forefront of technological innovation within the language-learning app sector. It demonstrated the app's commitment to using innovative technology to enhance educational outcomes, appealing to tech-savvy learners, and differentiating Translatinator from competitors.
    - Provided Continuous Value: Continuous updates and expansions to the object recognition feature, such as adding more objects beyond cats and dogs, ensured that the app remains valuable and interesting to users over time. This ongoing development signaled to users that their investment in the app continues to yield new learning opportunities and features.
    - Improved Learning Outcomes: The feature supported a more intuitive and natural way of learning languages, mimicking the way individuals learned their first language through direct interaction with the world around them. This can help improve vocabulary retention and comprehension, enhancing overall learning outcomes.

## Data Summary

The source of the raw data for the Translatinator enhancement project involved combining two Kaggle data repositories containing images of cats and dogs. These images were categorized to train a machine learning model capable of recognizing these animals within the application. This approach allowed for the creation of a robust dataset necessary for the development of an effective object recognition feature.

* Data Collection and Preparation
  + Source Identification: The raw data was sourced from two distinct Kaggle repositories, which provided a diverse collection of cat and dog images. The URLs to the repositories are linked here:
    - <https://www.kaggle.com/datasets/alifrahman/dataset-for-wbc-classification>
    - <https://www.kaggle.com/datasets/samuelcortinhas/cats-and-dogs-image-classification>
  + Categorization: After sourcing, the images were categorized into 'cats' and 'dogs' to facilitate supervised learning. This step involved organizing the images into structured directories or labels that the model could use during the training process.
* Data Processing and Management Across the Development Lifecycle
  + Design Phase
    - Data Analysis: Initially, the total image count was determined, establishing the dataset's size and variety. This step was crucial for planning the model's complexity and training requirements.
    - Preprocessing Setup: The data was then prepared for training through resizing and normalization to ensure uniformity. This included setting a consistent image height and width and rescaling pixel values for optimal model performance.
  + Development Phase
    - Training and Validation Split: The dataset was split into training and validation subsets, with 20% of the data reserved for validation. This separation is vital for evaluating the model's performance on unseen data.
    - Data Augmentation: To enhance the model's ability to generalize and prevent overfitting, data augmentation techniques such as random flipping, rotation, and zoom were applied. This process artificially expanded the training dataset with modified versions of the original images.
  + Maintenance Phase
    - Continuous Monitoring and Updating: After deployment, the model's performance was continuously monitored, with adjustments made based on real-world feedback and evolving data requirements. This included potential re-training or refinement to maintain or improve accuracy and user satisfaction.
    - Model Saving and Documentation: The final model was saved and documented, ensuring that the development process, model architecture, and training parameters were well-recorded for future reference or enhancement.

Throughout these phases, TensorFlow and Keras libraries were utilized for model development, demonstrating the application of industry-standard tools for machine learning projects. This systematic approach to data collection, processing, and management ensured that the project remained aligned with its objectives, from enhancing the Translatinator app's functionality to ensuring the model's adaptability and long-term utility.

## Machine Learning

* The implementation of the machine learning solution for the Translatinator project involves several key methods, each critical for the development and performance of the object recognition feature. Below is an explanation of these methods, covering what they do, how they were developed, and why they were chosen.
* Method 1: Image Dataset Preprocessing
  + What: This method involves resizing images to a uniform dimension (180x180 pixels) and rescaling pixel values to a [0,1] range.
  + How: TensorFlow's utils module was used to create image datasets from the directory, automatically handling the resizing and splitting into training and validation datasets. The layers.Rescaling layer was then applied to normalize pixel values.
  + Why: Uniform image sizes and normalized pixel values are essential for training a machine learning model efficiently and effectively. This preprocessing step ensures that the model receives data in a consistent format, improving training speed and convergence.
* Method 2: Convolutional Neural Network (CNN) Model
  + What: A CNN model architecture was chosen for its ability to recognize visual patterns in images with high accuracy. The model includes convolutional layers, max pooling, and dense layers.
  + How: The Sequential model from TensorFlow's Keras API was used to stack layers. The model starts with a rescaling layer, followed by multiple sets of convolutional layers and max pooling layers, a flattening layer, and finally, dense layers for classification.
  + Why: CNNs are the standard in image recognition tasks due to their ability to capture hierarchical patterns in images. “Instead of preprocessing the data to derive features like textures and shapes, a CNN takes just the image's raw pixel data as input and ‘learns’ how to extract these features, and ultimately infer what object they constitute” (Google, n.d.-a). The architecture was developed to balance complexity and performance, ensuring accurate object recognition while maintaining manageable computational requirements.
* Method 3: Data Augmentation
  + What: Data augmentation artificially increases the diversity of the training dataset by applying random transformations (e.g., flipping, rotation, zooming) to the images.
  + How: A Sequential model containing data augmentation layers was applied to the training dataset. This model includes random flips, rotations, and zooms.
  + Why: Data augmentation prevents overfitting and improves the model's generalizability to new, unseen images. By exposing the model to various transformations of the training data, it learns to recognize cats and dogs in different orientations and scales, enhancing its robustness.
* Method 4: Model Training and Validation
  + What: The process of training the CNN model on the prepared datasets and evaluating its performance on a validation set.
  + How: Training was performed over multiple epochs, with performance metrics collected for each epoch. According to Google (n.d.-b), an epoch is when each piece of data has been processed once. In this case, a preliminary and final model were trained over 10 and 15 epochs, respectively.
  + Why: Training and validation are crucial for developing a model that not only learns the training data but can also generalize well to new data. This method ensures the model is accurate and effective for the object recognition task without overfitting to the training data. Overfitting occurs when your model matches the training data so closely that it has trouble making predictions on new data (Google, n.d.-b).
* Method 5: Model Evaluation and Prediction
  + What: After training, the model's ability to classify new images as either cats or dogs is evaluated using previously unseen images.
  + How: A new image must first undergo preprocessing to match the format and characteristics of the images used to train the model. This typically involves resizing the image to the specific dimensions expected by the model (e.g., 180x180 pixels). To interpret the model's scores as probabilities, the softmax function is applied to the output vector. The softmax function takes a list of values and outputs a set of classifications and percentages (all summing up to 100%), thus making them interpretable as probabilities that the image belongs to each of the possible classes (Bhaskhar, 2020).
  + Why: This final step validates the model's practical utility and effectiveness in real-world scenarios. Evaluating the model on new images ensures it has learned to accurately identify cats and dogs, fulfilling the project's goal of enhancing the Translatinator app with a useful and engaging feature.

## Validation

For each employed method described in the section above provide the following:

* Method 1: Image Dataset Preprocessing
  + Validation Method: Visual inspection of a subset of the preprocessed images alongside their original versions to ensure resizing and normalization have been applied correctly.
  + Results: The validation might reveal that all images have been uniformly resized to 180x180 pixels without distortion, and pixel values are correctly normalized to the [0,1] range. This confirms that the preprocessing pipeline is set up correctly and ready for model training.
* Method 2: Convolutional Neural Network (CNN) Model
  + Validation Method: Use a hold-out validation set (data not seen by the model during training) to evaluate model performance metrics such as accuracy, precision (how many predictions were correct), recall (how often the model correctly identified as the positive class), and F1 score (a formula combining precision and recall) (Google, n.d.-b).
  + Results: The CNN model could demonstrate high accuracy and F1 scores on the validation set, indicating it has effectively learned to recognize visual patterns in the images. For instance, an accuracy of 95% and a similar F1 score would suggest the model is highly effective in classifying cats and dogs from images.
* Method 3: Data Augmentation
  + Validation Method: Training two models - one with data augmentation and one without - on the same training set and comparing their performance on a validation set to assess the impact of data augmentation on model generalizability.
  + Results: The model trained with data augmentation might show improved performance on the validation set, demonstrating lower overfitting compared to the model trained without augmentation. For example, the data-augmented model could achieve a 5% higher accuracy on the validation set, validating the effectiveness of augmentation in enhancing model robustness.
* Method 4: Model Training and Validation
  + Validation Method: Monitoring the learning curves (training and validation loss and accuracy over epochs) to assess model learning progress and detect overfitting.
  + Results: Ideally, both training and validation loss (how far a model’s prediction is from its label) decrease over time, with validation loss closely tracking training loss, indicating good model generalization (Google, n.d.-b). A divergence, where training loss continues to decrease but validation loss increases, would signal overfitting. Successful validation would show converging training and validation loss curves by the final epoch, e.g., training loss at 0.05 and validation loss at 0.06.
* Method 5: Model Evaluation and Prediction
  + Validation Method: Conducting a quantitative evaluation using a test set of images not used in training or validation. Metrics could include accuracy, as well as a confusion matrix to understand specific prediction errors.
* Results: The evaluation might reveal an accuracy of 94% on the test set, with the confusion matrix showing a higher misclassification rate between certain breeds or similar-looking animals, indicating areas for future model improvement. The high overall accuracy confirms the model's effectiveness in classifying new images as either cats or dogs.

## Visualizations

* Inside the zipped code file will be a folder named “required\_images”. This contains 3 images:
  + pre\_transformation.png: This is a set of charts that shows the training and validation accuracy before image manipulation/transformation occurs.
  + transformation\_example.png: This is an example showing the kinds of transformations that were done to increase the accuracy of the predictions.
  + post\_transformation.png: This is a set of charts that shows the training and validation accuracy after the transformations occurred.

## User Guide

* Visit this URL for the user guide: <https://github.com/asteurer/capstone/blob/main/README.md>

# Reference Page

* Google. (n.d.-a). Introducing Convolutional Neural Networks. Retrieved from <https://developers.google.com/machine-learning/practica/image-classification/convolutional-neural-networks>
* Google. (n.d.-b). Machine Learning Glossary. Retrieved from <https://developers.google.com/machine-learning/glossary>
* Bhaskhar, N. (2020). Interpreting logits: Sigmoid vs Softmax. Retrieved from <https://web.stanford.edu/~nanbhas/blog/sigmoid-softmax/>