FoodVideoQA: A Novel Framework for Dietary Monitoring

Krish Shah[1]*, Siddharth Viswanath[1]*, Pengcheng Xi[2], Chris Czarnecki[3]†, Yuhao Chen[3]†
[1]University of Waterloo
[2]National Research Council Canada
[3]Vision and Image Processing Group, Systems Design Engineering, University of Waterloo
{k33shah,snviswan,cczarnecki,y2863che}@uwaterloo.ca
pengcheng.xi@nrc-cnrc.gc.ca

January 23, 2025

Abstract

Food intake monitoring is a crucial area of research in food computing due to its complexity and significant potential for improving health outcomes. While traditional 2D image-based dietary assessments provide basic information, video offers a more detailed understanding of both the quantity of food consumed and the manner in which it is eaten. However, current video-based dietary analysis remains limited to coarse metrics, such as counting bites. In this paper, we introduce FoodVideoQA, a novel approach that leverages Vision-Language Models (VLMs) to analyze food intake videos comprehensively. Our framework includes lists of ingredients, utensils, consumed foods, and specific time intervals in a video where a person is eating. This work paves the way for more advanced multimodal food intake measurement and behavioral studies.

1 Introduction

In the world of healthcare, monitoring food intake is pivotal in forming dietary routines, especially for sensitive populations such as children and the elderly [1, 2]. Tracking dietary intake has been proven to prevent disorders such as malnutrition, diabetes, and cognitive decline for elderly people in nursing homes and assisted living [3, 4]. In nurseries, food tracking facilitates shaping healthy dietary habits at a young age, while preventing long-term conditions such as obesity [5]. Such applications require the identification of food items being consumed, a thorough nutritional breakdown, and an analysis of their ingredients.

To tackle the problem of dietary monitoring, many model architectures have been proposed, requiring extensive training or fine-tuning existing models [6, 7, 8, 9]. This approach is inflexible and is susceptible to domain changes. For example, if a model has been trained to count the number of bites of food taken, a task requiring the amount of time for which the food was eaten would require starting from scratch altogether. An additional downside of this approach is that it requires training data to be assembled that captures the various ways a person can eat. This is inherently limiting, as there are countless ways a person can eat, such as by food or by utensil. Since it is not practically feasible to incorporate every possible combination of food and utensils into a dataset, the training data is hence limited to a finite number of food and utensils [6, 7, 8, 9]. As a result, a model likely would not understand a person eating in real-world in cases where the food and utensil combination is not covered by the training data. These models are overly specialized to their training data, thus limiting their ability to perform well on real-world data.

Many previous approaches to dietary monitoring involve the use of wearable devices [10, 11]. The weakness of this approach is that it requires users to consistently wear these devices. Using Vision-Language Models (VLMs) allows us to automate inferences from video input and does not require deliberate action on the part of the user in their day-to-day lives.

To address these shortcomings, we build our method on Vision-Language Models, which can help perform food analysis with minimal training. Using foundational VLMs helps address this problem of generalization. An additional upside of this approach is that fewer resources are needed as expensive GPU training is not necessary.

Existing VideoQA methods struggle with real-world content since the datasets they use to train on are mainly

*Authors contributed equally
†Corresponding author

focused on shorter videos, typically under a minute. As such, VLMs have difficulty with memory requirements of time-based coherence over extended video sequences [12]. This is why we split videos into frames and focus on analyzing consecutive frames, rather than processing the entire video directly. This enables us to identify both the appearance and disappearance of individual food articles, while incorporating temporal context into our VLM-based analysis [13, 14].

Building on our VLM analysis, another important element we take into consideration is the subject's actions in the video – particularly, what they *do* with the food. The mere presence or absence of a food item in a single frame does not indicate its consumption. For example, a food item might temporarily disappear in a frame due to occlusion: such as lifting a spoon of soup to the mouth, where the food may momentarily be blocked by the hand or utensil. To address this, we incorporate mouth detection and food localization using pose estimation to better capture eating actions [15].

In this study, we present a framework that combines VLMs and Pose Estimation for a detailed assessment of a subject's food intake in videos. VLMs provide contextual descriptions of visible food items, their corresponding ingredients, and utensils. DWPose [15] tracks the subject's mouth landmarks, and GroundingDINO [16, 17] draws food bounding boxes – which helps us detect eating actions by measuring mouth openness and calculating the proximity of the nearest food item. Our experiments validate this framework's ability to label each frame as "eating" vs. "not eating", and identify the food item being consumed in the frame – with all labeled frames compiled into a final video with intervals. Our approach offers a scalable solution for dietary behavioral analysis.

2 Methods

Our approach makes use of Vision-Language Models (VLMs) and pose estimation to analyze food intake video data. We first use VLMs to identify nutritional content, ingredients, and utensils in each frame of the video to give us contextual information about visible food items.

Then, to detect whether a subject is actively eating, we leverage pose estimation. We focus on the subject's mouth positioning and openness while localizing food items near the mouth. This workflow combining VLM insights in and pose estimation in Figure 1 gives us a thorough assessment of eating behavior.

2.1 VLM-Driven Insights

We start by employing a VLM to extract information on nutritional values, ingredients, and utensils from each frame of a video. After parsing VLM outputs for each frame, we identify intervals of "consistent" eating behavior by grouping consecutive frames where the same food item is present. This ensures that our analysis occurs over short frame sequences rather than haphazard, isolated frames. Our approach aims to balance efficiency with high performance, and the overall workflow is depicted in Figure 1.

2.1.1 Individual Frame Analysis

Analyzing every single frame in a video for insights is inefficient and redundant. As frames separated by small timeframes will have minimal differences between them, it makes sense to pick frames that are reasonably spread out in time to observe significant changes. Frames are sampled from the video at regular intervals, τ, which represents the number of frames to skip between samples to extract. The value of τ is showcased in Table 1.

We use VLMs to generate insights for each image, by passing in a series of associated prompts. Our prompts elicit detailed descriptions of nutritional information – including food type, ingredients, utensils, and nutrient composition, showcased in Figure 2.

2.1.2 Interval Generation

To track food consumption over time, we analyze descriptions generated by the VLM for pairs of adjacent frames. Specifically, we parse the VLM outputs to isolate food items present in each frame and store them as an associated list with the frame. Next, we implement an interval-detection algorithm to identify consecutive frames that contain the same food item and mark the start and end of the interval. This allows us to quantify the appearance and disappearance of food items and provides a more detailed assessment of food intake on the frame level.

We also use a frame tolerance threshold, ϵ, to handle brief absences of a food item between frames. The value of ϵ is a hyperparameter detailed in Table 1. This added measure allows us to distinguish between actual changes in food consumption versus minor fluctuations in the VLM outputs. It helps our framework account for brief detection lapses, like occasion or brief misdetections, without mistakenly concluding that a food item has been removed.

By integrating this tolerance, our interval-detection algorithm more accurately represents continuous or intermittent consumption across consecutive frames.

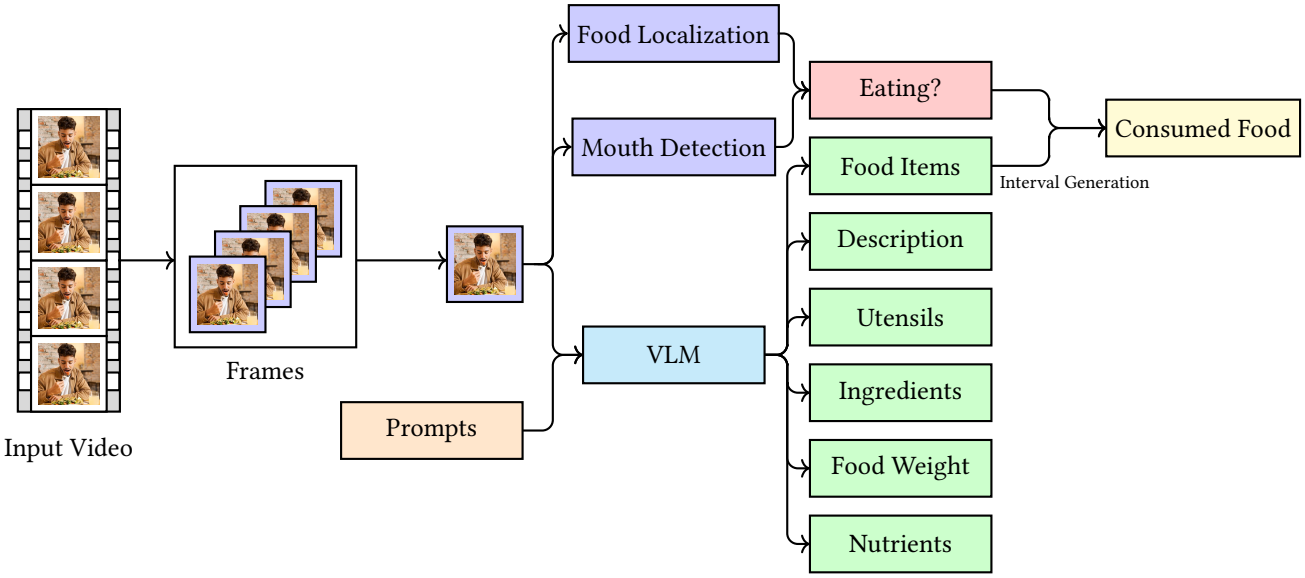

Figure 1: FoodVideoQA workflow diagram combining VLMs and Pose Estimation for food intake assessment given a video input. Video is processed frame-by-frame, VLM identifies visible food, ingredients, utensils, food weight, and nutrient information; interval-detection algorithm 1 analyzes adjacent frames to identify food items consumed in frame intervals; and Mouth detection and Food Localization via Pose Estimation determines whether the subject is eating.

Note: The pseudocode for Algorithm 1 showcases a naive implementation of the interval-generation algorithm, without the frame tolerance.

Algorithm 1 Interval Generation using VLM Output

Require: $food\_data$: List of detected food items per frame

Initialize an empty list $intervals$
Set the starting frame index $curr\_idx$ to 0
while there are frames left in $food\_data$ **do**
  Skip frames with no detected food items
  For each food item in the current frame:
  - Identify the next frame where this item reappears.
  - Record intervals only if they span at least two consecutive frames.
  - Store each valid interval along with its length.

  Sort potential intervals by length, prioritizing longer intervals
  Select the longest interval for each detected item in this frame and add it to $intervals$
  Advance $curr\_idx$ to the end of the selected interval to avoid overlapping intervals
end while
return List of intervals representing consistent eating periods for each food item. =0

2.1.3 Quantitative Validation

To measure the semantic accuracy of the VLM's output against our ground truth for ingredients and utensils, a simple word-to-word comparison is insufficient. Due to the variability of VLMs, there can be slight differences in interpretation between the VLM's output and the ground truth.

Specifically, semantic variation and visual similarity cannot be captured by one-to-one matching. For example, "cilantro" and "coriander" are two distinct words that mean the same herb, illustrating how language can vary while representing the same item. Visual similarity refers to the idea that two different objects can appear alike and may be easily mistaken for one another, such as an orange and a grapefruit. Although the VLM is not technically wrong in these cases, one-to-one matching would not recognize the semantic similarity.

To address this issue, we seek a method to quantify the semantic accuracy of the ingredient and utensil lists identified by the VLM. We adopt the BERTScore metric proposed by Zhang et al. [18] as a solution. This method provides a revised F1 score that is adjusted for the context of semantic matching. We use the *bert-score* Python package that implements the paper for this purpose.

During our experimentation, we found notable differences between the performance of one-to-word matching in comparison to semantic matching. Using Scikit-learn [19], we found that the average F1 score for one-to-one word matching was 0.31 in comparison to 0.78 for semantic matching, in a test run over 100 sample lists.

As an illustrative example, consider the following scenario:

VLM-generated List:
`['pasta', 'meatballs', 'tomato sauce', 'parmesan']`
Ground Truth List:
`['spaghetti', 'meatball', 'marinara', 'cheese']`

F1 Score (One-to-One Matching): **0.25**
F1 Score (Semantic Matching): **0.65**

One-to-one matching seems to only match exact terms, such as "meatball" and "meatballs". Whereas, Semantic matching via BERTScore performs more effectively by recognizing the semantic similarity between words in both lists. For example, parmesan is a type of cheese; spaghetti and pasta are similar items; and marinara resembles tomato sauce.

2.2 Pose Estimation

The second component of our video analysis framework is centered around pose estimation. We identified human pose as crucial information to be extracted for determining whether or not a person is consuming food in a single frame. This validation involves two separate conditions. The first condition checks whether the person's mouth is open, and the second condition verifies if there is a food item located near the person's mouth. A person is considered to be eating in a given frame if both conditions are met.

2.2.1 Tracking Lip Landmarks

While the process of checking if a person's mouth is open could be achieved through a neural network, we propose a simpler heuristic that leverages the DWPose [15] library to generate facial landmarks for this purpose.

DWPose generates facial and lip landmarks that allow us to determine mouth openness, shown in Figure 3. We analyze these landmarks to decide if the mouth is open, by defining "openness" based on the average distance between the upper and lower lip keypoints. In particular, we define the mouth to be open if and only if the average distance between the two lips is greater than the lip separation threshold β, whose value is showcased in Table 1.

Let $\vec{t}$ represent the points y-coordinates of the top of the lip and $\vec{b}$ represent the points representing the y-coordinates of the bottom of the lip. We average the distance across 3 corresponding points in the top and bottom of the lip.

Defining a boolean variable ξ to be 1 when a person is eating, and 0 when they are not, we check that this average distance is greater than or equal to a lip separation.

Figure 2:
Sample frame fed into VLM, along with associated prompts and responses as shown in below listing.

```
prompt: Identify only the food items visible in the
image. Provide a comma-separated list of food items
with no additional descriptions or details. Do not
repeat any items in your response.
answer: chicken, fried chicken, chicken wings

prompt: Provide a list of cutlery/utensils that the
person in the image is eating with, from this list:
[spoon, fork, knife, chopstick, spork, ladle, tongs,
spatula, straw, bowl, cup, glass]. Only provide a
comma-separated list of items with no additional
descriptions for each item in your response.
answer: cup

prompt: Provide a detailed list of the ingredients
of the food in the image. Only include a
comma-separated list of items with no additional
descriptions for each item in your response.
answer: chicken, breading, seasoning

prompt: Provide nutritional value about the food
you see in the image in bullet point format with
JUST this information and nothing else:
- Calories = ?
- Fats = ?%
- Protein = ?%
- Carbohydrates = ?%
answer:
- Calories: 1200
- Fats: 50%
- Protein: 20%
- Carbohydrates: 30%
```

The evaluation corresponds to the following:

$$\xi = \begin{cases} 1 & \left(\frac{\sum_{i=1}^{3}(t_i - b_i)}{3}\right) \geq \beta \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

2.2.2 Food Localization

Knowing if a person's mouth is open is not sufficient information to indicate eating – food must also be near the mouth. An open mouth could simply indicate someone is speaking, so the key factor is the proximity of food to the mouth. Assessing this condition is the focus of food localization.

First, we need to identify the locations of a person's mouth as well as food items, in a given frame.

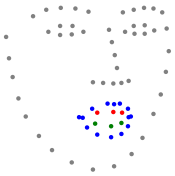

Figure 3: A face plot drawn using Matplotlib [20], outlining mouth in blue, top of lip in red, bottom in green.

We accomplish this via localization, using the GroundingDINO model [16, 17] to generate relevant bounding boxes for the text labels "food". We capture mouth details via DWPose – by extracting lip landmarks, and then drawing a bounding box around the mouth. Using DWPose helps ensure that we still detect the mouth even if partially occluded, for example, by food like corn on the cob.

Tightening the condition on the presence of food, we further require the food to be close to the mouth. An example is showcased in Figure 4. This closeness is quantified by the *intersection-over-union* (IoU) of food bounding boxes with the bounding box covering a person's mouth. A food item is defined as "in range" of the mouth if it meets an IoU threshold, δ:

$$\frac{F \cap M}{F \cup M} \geq \delta \tag{2}$$

where F and M represent the bounding boxes for the food and mouth in $\mathbb{R}^2$. The value of the hyperparameter δ is showcased in Table 1. When multiple food bounding boxes are detected, the closest one to the person's mouth is selected for testing against the threshold. If no food bounding boxes are identified, the condition will automatically fail.

3 Experimental Results

Table 1: Hyperparameters Used in Experiments

Hyperparameter	**Symbol**	**Value**
Frame Step Size	τ	20 frames
Frame Tolerance Threshold	ϵ	15 frames
Lip Separation Threshold	β	8.0
IoU Threshold	δ	0.15

We first collected a dataset of 10 videos that contain people from different demographics, eating a wide variety of food items. Each video contains approximately 100 consecutive frames, each manually labeled by us and

Mouth open: True — IoU: True — Eating: True

Mouth open: True — IoU: False — Eating: False

Figure 4: A comparison of eating vs. not eating states based on our pose estimation framework.

extracted based on our frame step size τ detailed in Table 1. We've also included videos of people with no food present in the video to ensure diversity in our dataset. We name our dataset NutriQuest. The dataset and our approach can be found on this GitHub repository.

Table 2: Comparison of VLM Performance

Model Name	$F1_{BERT}$	P_{BERT}	R_{BERT}
LLaVA-v1.6-7b	**0.66**	**0.60**	**0.71**
LLaVA-v1.5-7b	0.36	0.33	0.38
Blip-2 (LAVIS)	0.29	0.23	0.36

The selection of LLaVA as the VLM to solve our problem was based on its superior performance in comparison to VLMs showcased in Table 2. We tested LLaVA on our NutriQuest dataset with BERTScore ($F1_{BERT}$) as our metric, and we found that LLaVA consistently demonstrates a high BERTScore. Note that BERTScore assigns a default score of 0.0 to any instance where a list is empty, be it VLM-generated, or ground-truth. To account for this discrepancy, we assign a score of 1.0 to the case where both lists are empty. Table 2 presents the average score across all evaluated frames, where a high score indicates better performance.

We applied our Pose Estimation algorithm to the Nu-

Table 3: Pose Estimation Accuracy

Method	Accuracy
Combined	**0.87**
Food Localization (IoU)	0.81
Facial Landmarks (Mouth Open)	0.67

triQuest dataset and conducted ablation studies to evaluate the contribution of each component. Table 3 showcases the accuracy of our Pose Estimation algorithm on the NutriQuest dataset – where we tested the food localization and facial landmark detection separately, and then evaluated the combined approach. We see that a higher accuracy is achieved by our algorithm that combines both detection and localization.

4 Conclusion

In our paper, we brought to light an automated approach to nutritional tracking with minimal training. We elaborated on FoodVideoQA, a two-stage process combining VLM-based semantic insights with pose estimation to estimate the quantity of food consumed by a person in a video. We showcased our tool's performance on a dataset of 10 videos, annotated with ground truth food items. FoodVideoQA is the initial step toward more advanced multimodal food intake measurements and other nutrition-tracking-related fields.

5 Acknowledgements

This work was supported by the National Research Council Canada (NRC) through the Aging in Place (AiP) Challenge Program, project number AiP-006.

References

[1] M. Kheirmandparizi, J.-P. Gouin, C. C. Bouchaud, M. Kebbe, C. Bergeron, R. Madani Civi, R. E. Rhodes, B.-C. Farnesi, N. Bouguila, A. I. Conklin, S. A. Lear, and T. R. Cohen, "Perceptions of self-monitoring dietary intake according to a plate-based approach: A qualitative study," *PLOS ONE*, vol. 18, no. 11, p. e0294652, Nov. 2023. [Online]. Available: https://dx.plos.org/10.1371/journal.pone.0294652

[2] L. J. Dominguez, N. Veronese, E. Baiamonte, M. Guarrera, A. Parisi, C. Ruffolo, F. Tagliaferri, and M. Barbagallo, "Healthy Aging and Dietary Patterns," *Nutrients*, vol. 14, no. 4, p. 889, Feb. 2022. [Online]. Available: https://www.mdpi.com/2072-6643/14/4/889

[3] M. Takemoto, T. M. Manini, D. E. Rosenberg, A. Lazar, Z. Z. Zlatar, S. K. Das, and J. Kerr, "Diet and Activity Assessments and Interventions Using Technology in Older Adults," *American Journal of Preventive Medicine*, vol. 55, no. 4, pp. e105–e115, Oct. 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0749379718319986

[4] N. Pidrafita-Páez, J. Silveira, E. Pinto, L. Franco, M. Romero-Rodríguez, M. L. Vázquez-Odériz, and NUTRIAGE Study Group, "Dietary Adequacy in Older Adult Nursing Home Residents of the Northern Iberian Peninsula," *Nutrients*, vol. 16, no. 6, p. 798, Mar. 2024. [Online]. Available: https://www.mdpi.com/2072-6643/16/6/798

[5] L. L. Bellows, Y. Lou, R. Nelson, L. I. Reyes, R. C. Brown, N. Z. Mena, and R. E. Boles, "A Narrative Review of Dietary Assessment Tools for Preschool-Aged Children in the Home Environment," *Nutrients*, vol. 14, no. 22, p. 4793, Nov. 2022. [Online]. Available: https://www.mdpi.com/2072-6643/14/22/4793

[6] Z. Tang and A. Hoover, "A new video dataset for recognizing intake gestures in a cafeteria setting," in *2022 26th International Conference on Pattern Recognition (ICPR)*, 2022, pp. 4399–4405.

[7] C. Wang, T. S. Kumar, G. Markvoort, J. Caby, H. Hallez, and B. Vanrumste, "Eating activity monitoring in home environments using smartphone-based video recordings," in *2022 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2022, pp. 1–5.

[8] C. Wang, T. S. Kumar, W. De Raedt, G. Camps, H. Hallez, and B. Vanrumste, "Eat-radar: Continuous fine-grained intake gesture detection using fmcw radar and 3d temporal convolutional network with attention," *IEEE Journal of Biomedical and Health Informatics*, vol. 28, no. 2, pp. 1000–1011, 2024.

[9] J. Qiu, F. P.-W. Lo, S. Jiang, Y.-Y. Tsai, Y. Sun, and B. Lo, "Counting bites and recognizing consumed food from videos for passive dietary monitoring," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 5, pp. 1471–1482, 2021.

[10] M. Farooq and E. Sazonov, "A novel wearable device for food intake and physical activity recognition," *Sensors*, vol. 16, no. 7, 2016. [Online].

Available: https://www.mdpi.com/1424-8220/16/7/1067

[11] A. Doulah, T. Ghosh, D. Hossain, M. H. Imtiaz, and E. Sazonov, ""automatic ingestion monitor version 2" – a novel wearable device for automatic food intake detection and passive capture of food images," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 2, pp. 568–576, 2021.

[12] H. D. Xinyu Fang, Kangrui Mao, "Mmbench-video: A long-form multi-shot benchmark for holistic video understanding," *38th Conference on Neural Information Processing Systems (NeurIPS 2024) Track on Datasets and Benchmarks.*, pp. 1–27, 2024. [Online]. Available: https://arxiv.org/abs/2406.14515

[13] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *NeurIPS*, 2023.

[14] H. Liu, C. Li, Y. Li, and Y. J. Lee, "Improved baselines with visual instruction tuning," 2023.

[15] Z. Yang, A. Zeng, C. Yuan, and Y. Li, "Effective whole-body pose estimation with two-stages distillation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4210–4220.

[16] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.

[17] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[18] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," Feb. 2020, arXiv:1904.09675 [cs]. [Online]. Available: http://arxiv.org/abs/1904.09675

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[20] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.