Food scene segmentation using hue features

Group 1

January 20, 2023

Abstract

Food scene segmentation tremendously impacts the process of learning one's culture. There have been several techniques developed specifically to segment scenes. In this report, we introduce a machine learning model that takes input as colors and returns scene boundaries. We greatly emphasize solving issues produced during the development of this project, not on the model's architecture to gain minor extra performance.

1 Introduction

Street food and sidewalk culture have been around throughout Vietnam's historical development. Street food is a dynamic concept and practice which is adapting to the modern era[1], not least to social media with numerous blogs and YouTube videos devoted to favorite street food stalls and sellers[2]. According to CEOWORLD U.S magazine's 2019 survey[3], Ho Chi Minh was ranked 4th out of 50 best cities for food-obsessed travelers globally. The figure is reasonable due to the wide variety of flavors in Vietnamese street food and their easy accessibility. Street food also increases the income of the tourism section of the entire country and creates an incentive for tourists to come to that specific country. Development of countries usually follows a similar path, in that particular path some cultural aspects can no longer go onwards along with the country including street food culture, and Vietnam is taking its steps into a developing country. Since street food culture is gradually vanishing throughout the process of social development, several tourists from Asia countries visited Vietnam as nostalgia for their countries in the past. Globalization is rapidly occurring all over the world, and we desire to enrich and popularize our proud street food culture with the rest of the world. To achieve that goal, we collect street food review videos from Youtube, label and classify whether or not a frame contains food using only color features, then apply machine learning models to the video scene segmentation process to understand and make better use of the data and produce final results.

Most street food review videos consist of several off-topic frames before getting to food-contained segments. Searching proves to be inefficient to perform manually and that problem will likely gradually cool the user's mood down. Therefore, finding the correct frames that contain audiences' interests has always been under concern. By segmenting food videos, we aim to help culture-learning audiences to focus on the learning process without spending time figuring out the valuable frames. In essence, to maximize users' experience. This work emphasizes online video streaming platforms such as Youtube, and Netflix,... and works as an extension for these platforms. The reason why we choose online video streaming platforms in general and Youtube in specific is people seem to tend spending more of their time watching food-related videos rather than reading some food review leaflets scattered on the street or stuck on the walls. Also, the length of a food-related Youtube video is another reason that is very appropriate for busy people with mainly short videos revolving around about 10 - 15 minutes duration. And the popularity of Youtube with the percentage of OTT watchers consuming content is 94.5%[4] making it the ideal platform for collecting data and extracting user comments and opinions. In the end, the final purpose of this extension is to help people to list food-contained segments and directly request selected segments from servers.

In this work, we only take food review videos recorded about Vietnamese street food as input to emphasize Vietnamese culture and the way people think about Vietnamese street food. Color features can easily be extracted as an array of lengths four (red, green, blue, and occupancy) in form of integers in the range 0 to 255. Because of the ease of processability and accessibility, color was chosen as the only input feature, our input includes 512 vectors with each scala representing the amount the color appeared in the frame. Our entire dataset contains 1034 videos from Youtube and is mainly about reviewing Vietnamese street food services.

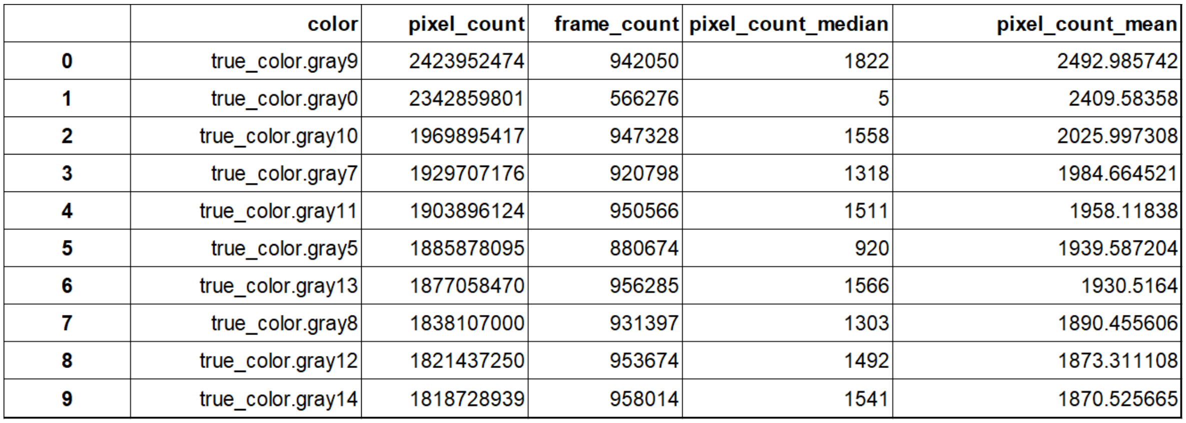
2 Related work

Within the past millennium, researchers have developed projects under the name of artificial intelligence(AI). However, AI itself consumes massive computational power that has never in used. In 2012, AlexNet[8] was the first deep convolution neural network(DCNN) architecture and trained with two NVIDIA GTX 580 3GB for approximately five days. Recent hardware breakthroughs combined with machine learning open-source platforms empowered massive deep neural network architectures that potentially surpassed AlexNet and unlocked the machine's unknown abilities. Several heuristically solved problems using machine learning can now be reimplemented by adapting deep learning. By having a deep learning approach, video scene segmentation is reaching new stages with better performance.

At the beginning of 21st century, video scene segmentation methods were not simple. H.Sundaram et al split scenes by detecting changes in audio and image motions[9], both stages involved complex mathematic formulas. The methods are inappropriate at present due to the lack of transparency, the complexity of implementation, and the length of processing time. In modern days, countless areas use video scene segmentation techniques to help with detecting key scenes. Scene segmentation is the task of splitting a video into its various semantic components called scenes. Extensive research has been performed using various techniques and brought about remarkable results. M.Mascorro et al. have conducted research about criminal intention detection at the early stages of shoplifting cases using mainly video scene segmentation technique and 3D CNN architecture[6]. The authors investigated suspicious scenes and collected previously related scenes. Then, the model was fed with these scenes and classified whether suspicious actions (shoplifting) exist. The work is extensively heavy to compute since several stages are involved and most of them are computationally expensive. Another work that involves in scene segmentation approach by Li et al. in traffic scenes with their proposed network and the help of deep learning[7]. The paper used RGB-D images as input and then compared the results with the method that only takes RGB images as input. Their proposed network architecture achieved good real-time performance and competitive segmentation accuracy. Hyesung et al[10] segment video scenes using Deep-learning Semantic-based Scene-segmentation model. The authors first detect shot boundaries, extract the keyframe and then feed the frame into DNN to generate image captions. Subsequently, query similarities scores were calculated and decided wether the scene is appropriate to the user's need. Applied methods involve long short-term memory neural networks (LSTM) - a common type of neural network specialized to train sequential data. LSTM was used to generate captions of the shot and a pre-trained model was taken. The author claims that it was unfeasible to train such a complex model from scratch. Trojahn et al.[11] created a combination of CNN and RNN with many meaningful features. Authors extract 4 features directly from the video and then feed them to LSTM models to generate scene boundaries. The work is generally more intensive than the abovementioned works and metrics are unclear. In conclusion, scene segmentation is an active field of research and currently attractive to researchers. According to our search term, most paper currently solve scene segmentation using deep learning based approaches because of amazing auto-corrected ability neural network offer.

In this paper, unlike other approaches, we use only colour features to yield a result of segmenting food scenes. This creates huge challenges since previous works were taken in well-defined meaningful inputs such as audio, and the frame itself,... Since we only count the coloured pixels, there are no localised meanings and therefore, can not apply CNN approaches, unless, modifications are established. Our contributions are:

- We train deep learning based models with only color features and report the results. (Specifically, we explore sequentially adaptive models.)
- We suggest an appropriate metric to evaluate the model's performance. In previous papers, no standard metrics for evaluations was provided. We wish to bring to our research community a standard to evaluate scene segmentation performance.



	color	pixel_count	frame_count	pixel_count_median	pixel_count_mean
0	true_color.gray9	2423952474	942050	1822	2492.985742
1	true_color.gray0	2342859801	566276	5	2409.58358
2	true_color.gray10	1969895417	947328	1558	2025.997308
3	true_color.gray7	1929707176	920798	1318	1984.664521
4	true_color.gray11	1903896124	950566	1511	1958.11838
5	true_color.gray5	1885878095	880674	920	1939.587204
6	true_color.gray13	1877058470	956285	1566	1930.5164
7	true_color.gray8	1838107000	931397	1303	1890.455606
8	true_color.gray12	1821437250	953674	1492	1873.311108
9	true_color.gray14	1818728939	958014	1541	1870.525665

Table 1: Most appeared color throughout the dataset.

3 Data preparation

Dataset used in this paper contains nearly 762 Youtube videos on Vietnamese street foods and then extracted to get one frame for every second of the videos. The dataset is then mapped to a pre-trained model to collect color features of each frame, each containing a color palette of 512 colors. Each color is represented by a number of pixels that were counted in the frame by our pre-trained model. Needless to say, the sum of all 512 colors will be approximately the total number of pixels in the frame due to missing quantiles within the pre-trained model.

In total of nearly 972,000 frames in the dataset, up to 7 GB of memory for the JSON format, to be able to analyze and process this data, we used MongoDB to export the dataset to CSV format and then utilize Python Pandas to extract useful information about the data. With that procedure established, we further explore the dataset and obtained the top ten colors in the dataset along with their respected median and mean for each, as shown in the table 1.

As we can see in Table 1, the top ten colors were dominated by gray. This phenomenon can be explained by the fact that most of the video frames extracted will be filmed by a video camera or a camera phone which when focused on the main subject will turn the out-of-focus range blurry which will be identified as gray when applying the pre-trained model. Furthermore, most of the street food videos were shot at night and on the street, therefore, as a result, the main color of each frame will certainly be dominated by gray. Another hypothesis is that most videos were shot with low-quality cameras and produced unwanted noisy signals. Lastly, our dataset is right skew as median generally less than mean and there are multiples grey values reached whole-frame size explained as fading effects within videos.

The challenges begin when we have to transform data into z-scale in order to feed them as inputs for the downstream deep learning model. Due to hardware constraints, take advantage of Google Colab pro with large RAM to produce the scaled version of the dataset, which turn 1.2 GB to 10 GB in CSV format. Because of high variances, the data contains a huge number of approximately zeros and zeros scalar. We assume this would significantly hurt the performance so dimensionality reduction techniques such as PCA were applied, however, failed due to hardware limits. As a brief summary, we use scaled z-score data as inputs for our downstream model.

4 Methods and system design

Figure 1 illustrates the overall architecture of our proposed model, including four distinct stages. The first stage is data preparation that was discussed in the earlier. Next is food frame classification; it detects which frame is food or non-food. Third, these results of the third stage are used to analyze

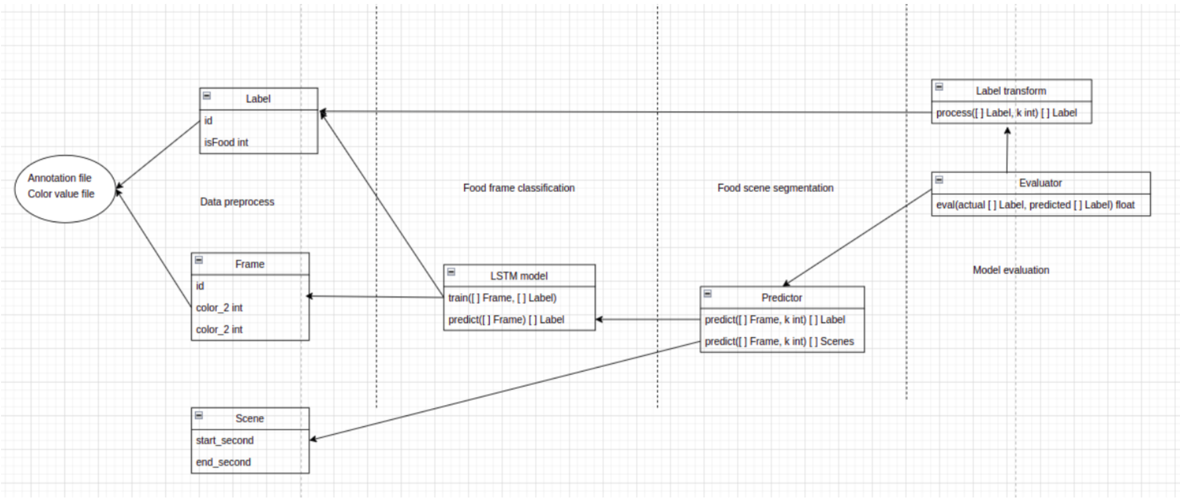

Label
id
isFood int
Label transform
process([ ] Label, k int) [ ] Label
Annotation file
Color value file
Data preprocess
Food frame classification
Food scene segmentation
Evaluator
eval(actual [ ] Label, predicted [ ] Label) float
Model evaluation
Frame
id
color_2 int
color_2 int
LSTM model
train([ ] Frame, [ ] Label)
predict([ ] Frame) [ ] Label
Predictor
predict([ ] Frame, k int) [ ] Label
predict([ ] Frame, k int) [ ] Scenes
Scene
start_second
end_second

Figure 1: Overall architecture.

and segment the video based on semantic elements and compare their similarities to each other. The final stage is scene assembly; the segmented scenes are used to evaluate the model , completing the proposed model.

4.1 Food frame classification.

Generally, a video is structured hierarchically consisting of shots, frames and scenes. Scenes are a continuous flow of frames, frames within a scene have a strong semantic correlation with each other. Therefore, food frame identification is the fundamental task in food-scene segmentation. For frame classification as food or non-food, we use the Long Short Term Memory-LSTM model. LSTM model is a variation of a recurrent neural network (RNN), which performs well in machine translation [12], speech recognition [13], sequence generation [14] and so on. LSTM model is composed of cells with multiple gates attached, and the cells behave in several ways, depending on the values of the gates that each cell is connected to. In short, the weights are trained on the same principle of how a value of a hidden layer is trained, and the learning process uses backpropagation through time to minimize the output error. The LSTM is a memory cell C that encodes the knowledge at all stages of the observed input up to the current stage"

The cell is controlled by the gate. In this layer, multiplication is applied; therefore, if the gate value is 1, then the gated layer's value is kept, but if the gate value is 0, then the gated layer's value becomes 0. Three gates are being used to control whether to forget the current cell value (forget gate f), if it should read its input (input gate i), or whether to output the new cell value (output gate o). The LSTM is trained to predict each frame output based on the hidden state of the previous frame sequence. Once the training phase is finished, the LSTM generates a series of predicted frames of the video, with 1 being food and 0 being non-food. The output of this stage is then used to segment the videos.

4.2 Scene assembly

On this stage, we finally assemble the frames into scenes, completing our scene-segmentation process. To do this process, we have to solve an important problem. What happens when one or multiple food frames stand in between non-food frames, or there are one or multiple non-food frames that are in the food scene, do we split the scene into 2 different scenes or do we ignore the non-food frames. To tackle this, we use an algorithm that groups multiple frames together and employs a connector variable called k to define when to or not to group frames to a scene. This connector k functions as a threshold that helps the algorithms to decide when to group scenes. We applied this algorithm to both the training data and predicted data.

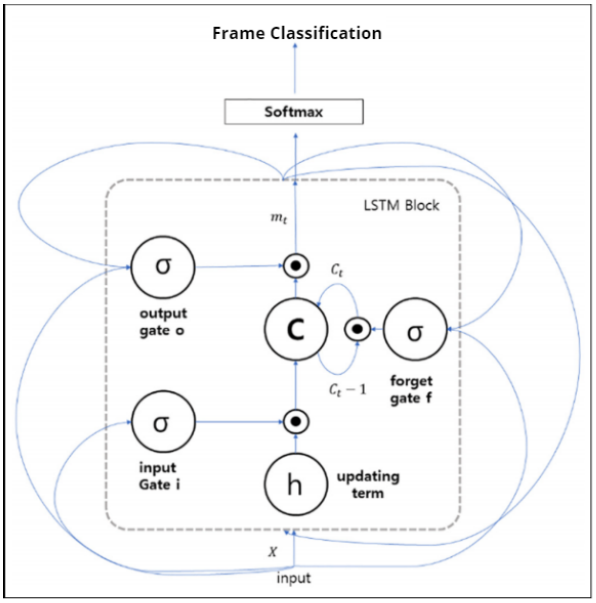


Figure 2: LSTM memory block.

To choose the most optimal k for this problem, we applied the brute-force method, by running the model through multiple iterations of k to choose the most appropriate answer. Based on our annotated dataset for street food Youtube video, the number of frames between two scenes normally falls in the range of [1-10]. Therefore, we decided to implement our scene assembly algorithm with k in the proposed range.

4.3 Evaluation metrics.

We evaluated the model based on two tasks: (1) accuracy of food-frame detection and (2) accuracy of semantic food-scene segmentation. First, food-frame detection were evaluated by model accuracy as shown in the equation below:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

Secondly, For calculating similarities between the extracted scenes and predicted scenes, we applied the Jaccard (IoU) similarity method:

$$JDist(A, B) = \frac{|A \ \cap B|}{|A \ \cup B|}$$

$$if\ A \cap B\ and\ A \cup B\ is\ \varnothing\ then\ J = 0$$

A is the frame-sequence result of our food-frame classification model using LSTM after it has been applied to the scene assembly algorithm. B is the sequence from our annotated dataset and applied to the assembly algorithm. The value becomes closer to 1 as the two become more similar. But if the predicted scene becomes more different, the value converges to 0.

The rationale behind the IoU score is that we only want to take into account class 1. As the long video might only contain a small volume of the food scene, we should not use accuracy as the model could fool us by mass producing outputs as 0 and still gain high performance. Instead, we can focus only on 1 label since that is the part where useful information exists. As we only take the predicted scenes and compare them to labeled scenes, we eliminate all irrelevant frames and successfully captured the model's useful outputs.

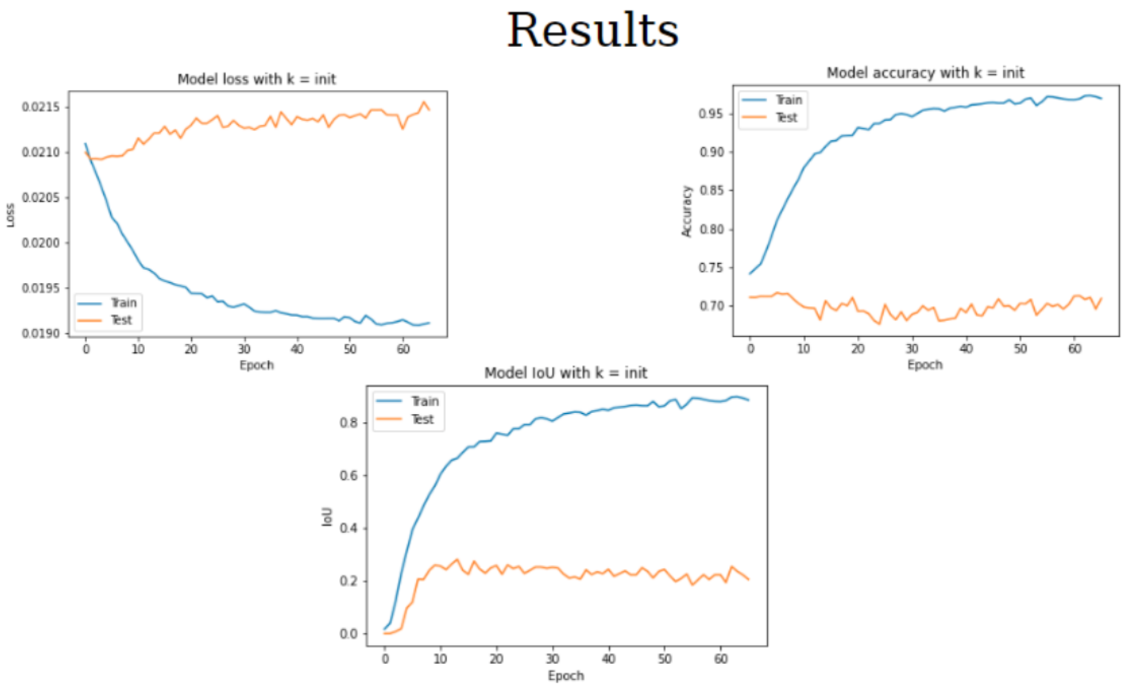


Figure 3: Model results with k = 5

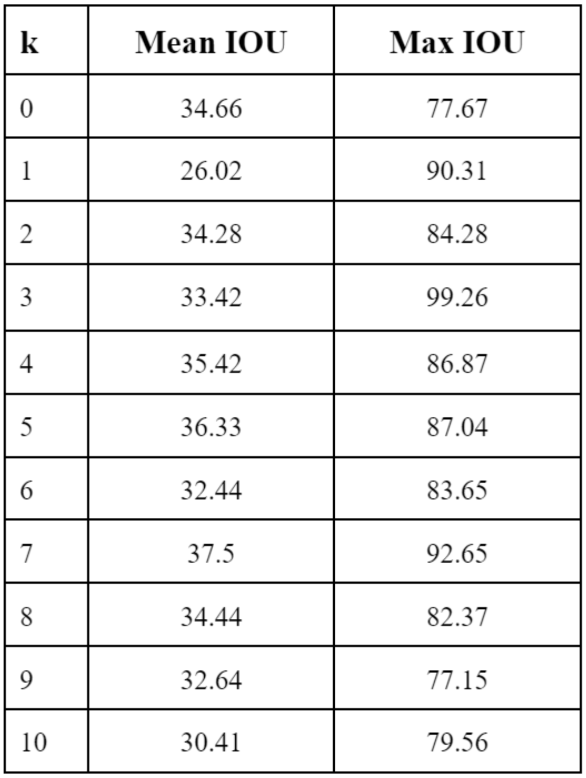
5 Results and Discussion

We tested several architectures to investigate how the model affected performance. We used grid search with these configurations:

- LSTM layers: [1,2,3,4,5]
- LSTM Hidden unit: [32,64,128]
- Activation on middle dense: ['relu','gelu','sigmoid','tanh']

And the architecture below yields the best result with 2 LSTM layers with hidden units equal to 64. Note that we tested these parameters with k=5. However, at the initial iteration, the result was terrible with an IoU score of approximately 10%. We double-checked the result and found that most of the frame's classification were below the threshold and therefore, rounded into 0. We suspected that the ratio between the number of food frames and non-food frames was greatly unbalanced and lowered the model's confidence about classifying the frame as containing food. We did a small research and decided to use focal loss as our alternative to cross entropy loss. Focal loss was introduced in RetinaNet as a solution to the label's unbalance problem. The loss function works by directly adjusting backward-gradient, specifically, reducing the importance of 0 class and significantly penalizing misclassified 1 class. With a more efficient loss function, we achieved a tripled result compared to the previous iteration. To save resources and improve iteration time for training and fine tuning , we first train the model to reach a stable accuracy, then save the weights of the stable model. After having a workable model, we train that model multiple times, each time with a different parameter k for scene segmentation. Each training session includes loading train/val data and initializing the model, if the weights are already saved as discussed above, we load the weights and compile our model. Then fit the mode with epochs equal 100, we then predict the test data to compute the IoU. After training. we have a mean IoU score for each k number. Parameter k was initialized to 5 for the dummy test, but we train the model for k in the range of [0,10] to validate the performance for each k.

Figure 3 shows that the loss value at k = 5 reached optimal around epoch 30-40 for the training set, and the loss of test set seems to remain the same throughout all epochs. For model accuracy, performance on the training set reached a maximum of nearly 97% and on the test set the accuracy plateaued around 70%, this is an impressive score for our food-scene classification with the LSTM



k	Mean IOU	Max IOU
0	34.66	77.67
1	26.02	90.31
2	34.28	84.28
3	33.42	99.26
4	35.42	86.87
5	36.33	87.04
6	32.44	83.65
7	37.5	92.65
8	34.44	82.37
9	32.64	77.15
10	30.41	79.56

Table 2: Model performance.

algorithm. For most food frames extracted from dataset videos, the dominant colors would be quite similar, which are light magenta which resembles the Youtuber's face when they are presenting the food. IoU value reached nearly 90% for the training set while converging around 25% for the test set. The reason might be that using only color feature may not sufficiently segment the video into scenes because of overlapping colors between food frame and non-food frame. The table below summarizes the results of our model for each iteration of k.

From table 2, it is clear that the best performance of scene segmentation is obtained when k = 7. The reason for this could be that with a lower value of k, the model had difficulty segmenting the scenes while a higher value of k the model tend to underfit and ignore the scenes.

5.1 Discussion

Even though the result was low in terms of IoU, we achieved a quite high accuracy at approximately 80% and successfully captured most food frames which suggests that our model could practically serve humans and we successfully achieved our forementioned goal. However, our maximum IoU among our test videos was remarkably high, so we made an investigation and came up with unproven hypotheses. First, videos with high color contrast tend to have higher IoU. As reported above, our dataset mainly contains gray colors and in the high IoU videos, less gray data points were found. Additionally, reviewers tend to stay in the same restaurant background and tell stories before the actual food scene. With those videos, our model reported an extremely bad result. We believe the problem violates the recursive nature of LSTM. Our color data does not change aggressively within shots, which does not provide margins for models to predict if within the shot contain food-scenes. The inputs to some aspects greatly limit our model's performance and that points out one important conclusion: We did not eliminate noise in the dataset well enough.

5.2 Future work

Firstly, our dataset was not cross validated or benchmarked with robust regulations. Reviewing the data and applying robust rules would clean the data even further. Further work includes defining more parameters to help the model better divide the video into scenes, for example, a parameter to define the threshold of minimum frames for a scene. There are approximately 49 scenes in the dataset containing less than 10 frames (equal to 10 seconds) which potentially hurt the performance. Even more, we could reduce input dimensions for better iteration and help the data itself be more compact.

Other metrics and architecture components can also help the model to better classify food-frames like the Micro F1 specifically for label 1. F1 score for label 1 is similar to IoU but captures precision and recall and therefore, yields a larger margin compare to IoU. Furthermore, applying bi-directional LSTM model for food frame classification has the potential to improve the performance of our model because unlike standard LSTM, the input flows in both directions, and it's capable of utilizing information from both sides. With bi-directional LSTM, the model can obtain shot's color changes that later occure which might fix the long-background problem mentioned above.

6 References

[1] Calloni, M. (2013). Street food on the move: A socio-philosophical approach. Journal of Science Food and Agriculture, 93, 3406–3413.
[2] Euromonitor. (2015). Q&A The resurgence of 'street food', 9 July. Retrieved from Euromonitor International database.
[3] Papadopoulos, A. (2019). The World's 50 Best Cities For Street Food-Obsessed Travellers from https://ceoworld.biz/2019/09/30/ranked-the-worlds-50-best-cities-for-street-food-obsessed-travellers-2019/
[4] GMI blog (2022) Youtube user statistics from https://www.globalmediainsight.com/blog/youtube-users-statistics/
[5] A. Mustafa, H. Kim, J. Guillemaut and A. Hilton, "Temporally Coherent 4D Reconstruction of Complex Dynamic Scenes," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4660-4669, doi: 10.1109/CVPR.2016.504.
[6] Martínez Mascorro, Guillermo & Abreu Pederzini, Jose Ricardo & Ortiz-Bayliss, José Carlos & Collantes, Angel & Terashima-Marín, Hugo. (2021). Criminal Intention Detection at Early Stages of Shoplifting Cases by Using 3D Convolutional Neural Networks. Computation. 9. 24. 10.3390/computation9020024.
[7] Li, Linhui & Qian, Bo & Lian, Jing & Zheng, Weina & Zhou, Yafu. (2017). Traffic Scene Segmentation Based on RGB-D Image and Deep Learning. IEEE Transactions on Intelligent Transportation Systems. PP. 1-6. 10.1109/TITS.2017.2724138.
[8] Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou & K. Q. Weinberger (ed.), Advances in Neural Information Processing Systems 25 (pp. 1097–1105) . Curran Associates, Inc. .
[9] Sundaram, H. & Chang, Fu. (2000). Video scene segmentation using video and audio features. 2. 1145 - 1148 vol.2. 10.1109/ICME.2000.871563.
[10] Ji, Hyesung; Hooshyar, Danial; Kim, Kuekyeng; Lim, Heuiseok (2018). A semantic-based video scene segmentation using a deep neural network. Journal of Information Science, (), 016555151881996–. doi:10.1177/0165551518819964
[11] Tiago Henrique Trojahn;Rudinei Goularte; (2021). Temporal video scene segmentation using deep-learning . Multimedia Tools and Applications, (), –. doi:10.1007/s11042-020-10450-2
[12]Cho K, van Merrienboer B, Gulcehre C et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation, 2014, https://arxiv.org/abs/1406.1078
[13] Graves A, Mohamed AR and Hinton G. Speech recognition with deep recurrent neural networks. In: 2013 IEEE international conference on acoustics, speech and signal processing (ICASSP), Vancouver, BC, Canada, 26–31 May 2013, pp. 6645–6649. New York: IEEE.
[14] Sutskever I, Vinyals O and Le QV. Sequence to sequence learning with neural networks. In: Advances in neural information processing systems, Montreal, QC, Canada, 08–13 December 2014, pp. 3104–3112. Cambridge, MA: MIT Press.

A Project management plan

A.1 Team members

Student ID	Full Name	Roles
SE160997	Nguyen Son Tung	Team leader
SE160753	Le Xuan Tung	Presenter
SE160880	Nguyen Ngoc Kha	Implementer
SE160466	Nguyen Cang Truong	Writer
SE160773	Nguyen Manh Tuong	Coordinator

A.2 Project

In this AI programming project, we will be focused on video scene segmentation using color for streetfood videos that have been captured in Vietnam. The objective of this project is to provide a comprehensive overview of using machine learning methods with color as main feature for food scenes segmentation and identification.

A.3 Planning

This project will be continously running for 10 weeks of the 2022 Summer Semester from start to finish. Devided to 6 phases as follows:

- Planning and project setup.
- Data collection and preprocessing: understand input and output of model in specific and project in general.
- Research related work: collect information of near field.
- Model Implementation: have a baseline model.
- Testing and evaluation: enhance model performance and correctness.
- Writing report & yield result: Final up project and submission.

Each phases of the project will take from 1 to 2 weeks to complete.

A.4 List of actions

Based on the project objectives and constrains, a version of Gantt chart are used to manage tasks. The project Schedule will be posted online and updated as tasks are completed. Any changes to the schedule must be documented in a revised project schedule. The detailed project schedule is accessible using Google docs and the whole project was managed using online tool ClickUp.

A.5 Risk management

As issues arise within the project, each member will determine if the issue is significant enough to report it to the team. The Team Lead, will decide if the issue should be reported to the full Workgroup. If so, the collaborative work site (Google docs site) will be used as a place to describe and track issues. For project work to continue efficiently, it is desirable that most issues be resolved with consultation with the Team Leader. Issues may include testing results, unexpected problems, and other items that impact project completion.