COMPSCI 682, Fall 24
Project Spotlights

Day 3 – Dec 10, 2024

Umass, Amherst

Fill out the SRTIs

Have until Thursday, December 19th, to complete the surveys

Link http://owl.umass.edu/partners/courseEvalSurvey/uma/

Tell us what you liked about the class, and what you didn’t.

Completely anonymous.

Instructions, once again

Speakers will have 2 mins to present their work

We will warn you at when 1 min, 30 sec, 0 sec remain

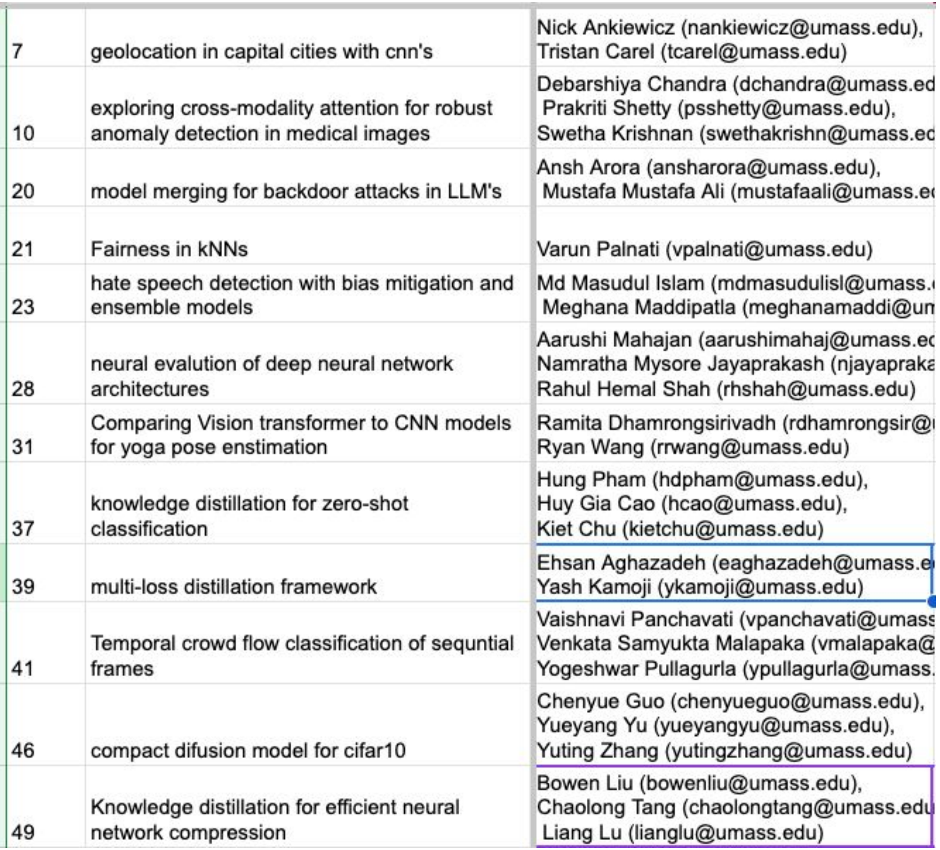
Must wrap up at 0

We will ask questions during grading

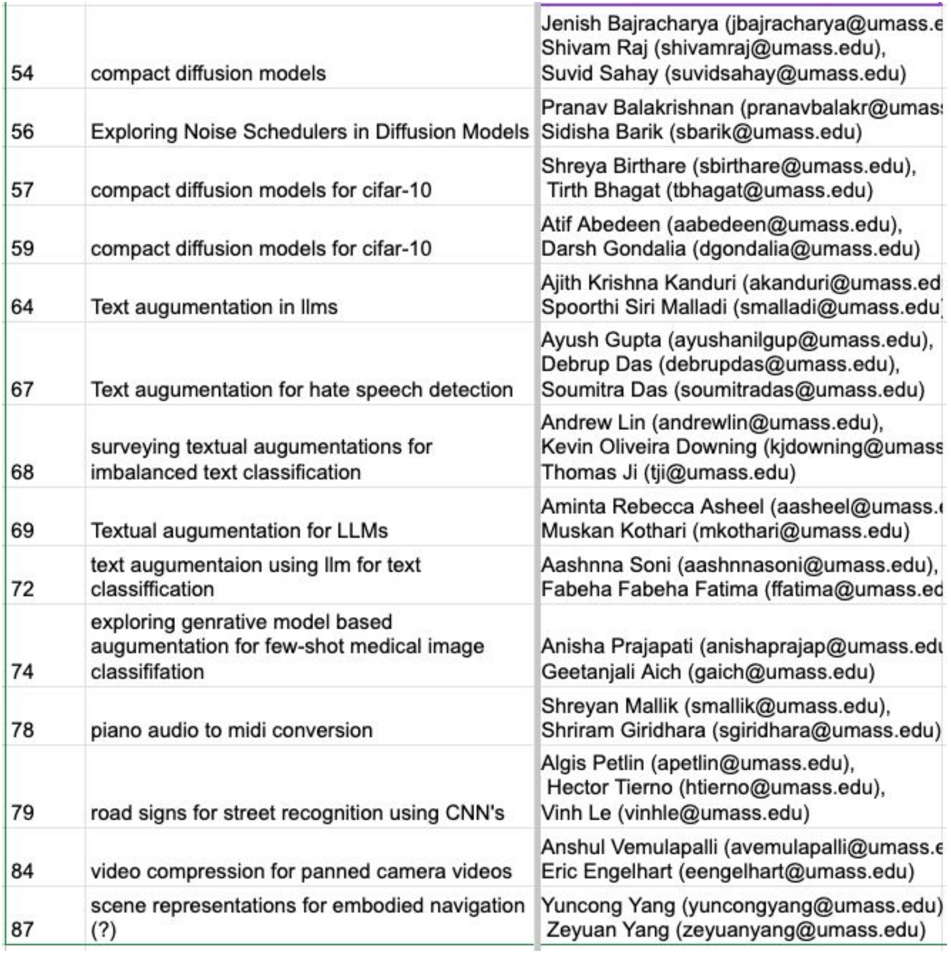
Thanks!

But first attendance

Presentation order



7	geolocation in capital cities with cnn's	Nick Ankiewicz (nankiewicz@umass.edu), Tristan Carel (tcarel@umass.edu)
10	exploring cross-modality attention for robust anomaly detection in medical images	Debarshiya Chandra (dchandra@umass.ed Prakriti Shetty (psshetty@umass.edu), Swetha Krishnan (swethakrishn@umass.ed
20	model merging for backdoor attacks in LLM's	Ansh Arora (ansharora@umass.edu), Mustafa Mustafa Ali (mustafaali@umass.e
21	Fairness in kNNs	Varun Palnati (vpalnati@umass.edu)
23	hate speech detection with bias mitigation and ensemble models	Md Masudul Islam (mdmasudulisl@umass. Meghana Maddipatla (meghanamaddi@un
28	neural evalution of deep neural network architectures	Aarushi Mahajan (aarushimahaj@umass.ec Namratha Mysore Jayaprakash (njayapraka Rahul Hemal Shah (rhshah@umass.edu)
31	Comparing Vision transformer to CNN models for yoga pose enstimation	Ramita Dhamrongsirivadh (rdhamrongsir@ Ryan Wang (rrwang@umass.edu)
37	knowledge distillation for zero-shot classification	Hung Pham (hdpham@umass.edu), Huy Gia Cao (hcao@umass.edu), Kiet Chu (kietchu@umass.edu)
39	multi-loss distillation framework	Ehsan Aghazadeh (eaghazadeh@umass.e Yash Kamoji (ykamoji@umass.edu)
41	Temporal crowd flow classification of sequntial frames	Vaishnavi Panchavati (vpanchavati@umass Venkata Samyukta Malapaka (vmalapaka@ Yogeshwar Pullagurla (ypullagurla@umass.
46	compact difusion model for cifar10	Chenyue Guo (chenyueguo@umass.edu), Yueyang Yu (yueyangyu@umass.edu), Yuting Zhang (yutingzhang@umass.edu)
49	Knowledge distillation for efficient neural network compression	Bowen Liu (bowenliu@umass.edu), Chaolong Tang (chaolongtang@umass.edu Liang Lu (lianglu@umass.edu)



54	compact diffusion models	Jenish Bajracharya (jbajracharya@umass.e Shivam Raj (shivamraj@umass.edu), Suvid Sahay (suvidsahay@umass.edu)
56	Exploring Noise Schedulers in Diffusion Models	Pranav Balakrishnan (pranavbalakr@umas Sidisha Barik (sbarik@umass.edu)
57	compact diffusion models for cifar-10	Shreya Birthare (sbirthare@umass.edu), Tirth Bhagat (tbhagat@umass.edu)
59	compact diffusion models for cifar-10	Atif Abedeen (aabedeen@umass.edu), Darsh Gondalia (dgondalia@umass.edu)
64	Text augumentation in llms	Ajith Krishna Kanduri (akanduri@umass.ed Spoorthi Siri Malladi (smalladi@umass.edu
67	Text augumentation for hate speech detection	Ayush Gupta (ayushanilgup@umass.edu), Debrup Das (debrupdas@umass.edu), Soumitra Das (soumitradas@umass.edu)
68	surveying textual augumentations for imbalanced text classification	Andrew Lin (andrewlin@umass.edu), Kevin Oliveira Downing (kjdowning@umass Thomas Ji (tji@umass.edu)
69	Textual augumentation for LLMs	Aminta Rebecca Asheel (aasheel@umass. Muskan Kothari (mkothari@umass.edu)
72	text augumentaion using llm for text classiffication	Aashnna Soni (aashnnasoni@umass.edu), Fabeha Fabeha Fatima (ffatima@umass.ed
74	exploring genrative model based augumentation for few-shot medical image classififation	Anisha Prajapati (anishaprajap@umass.edu Geetanjali Aich (gaich@umass.edu)
78	piano audio to midi conversion	Shreyan Mallik (smallik@umass.edu), Shriram Giridhara (sgiridhara@umass.edu)
79	road signs for street recognition using CNN's	Algis Petlin (apetlin@umass.edu), Hector Tierno (htierno@umass.edu), Vinh Le (vinhle@umass.edu)
84	video compression for panned camera videos	Anshul Vemulapalli (avemulapalli@umass.e Eric Engelhart (eengelhart@umass.edu)
87	scene representations for embodied navigation (?)	Yuncong Yang (yuncongyang@umass.edu) Zeyuan Yang (zeyuanyang@umass.edu)

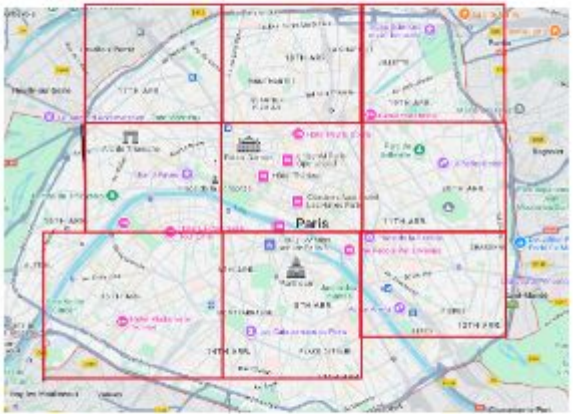
Geolocation in capital cities with CNNs

Nick Ankiewicz (nankiewicz@umass.edu),
Tristan Carel (tcarel@umass.edu)

Project #7

Summary

- Identifying the location of a still image is a significant challenge in computer vision, requiring a nuanced understanding of visual cues that convey geographic, architectural, and cultural information.
- Capital cities often serve as cultural and political hubs, featuring distinctive architectural styles, landmarks, and urban layouts that differentiate them from other regions
- We seek to know if capital cities contain distinctive enough features such that one can identify the city where a photo was taken based solely on the features within the image
- Previous works handle geolocation at a global scale, we chose to narrow it down to capital cities to see if they are distinctive enough to distinguish themselves from one another, and to see if there were any cities containing similar features (i.e. one city is commonly misclassified as another city)
- We treated this problem as a classification task, with one capital city per class. To make our dataset as comprehensive as possible, we divided our cities into several distinct areas (geocells), and sampled a number of images from each geocell based on multiple factors (size of cell, size of city, location of cell, etc.).



Division of Paris into 9 distinct geocells

Brussels

Rome

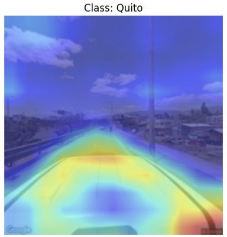
Stockholm

Washington DC

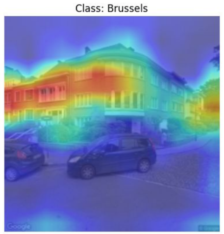
Examples of locations which our model was trained on

Main Results

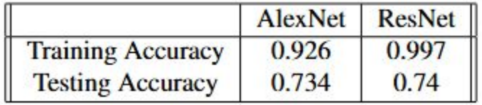
- Training both AlexNet and ResNet on our acquired dataset yielded test accuracies of .734 and .740 respectively
- Our accuracies fell between the other model’s city accuracies, and country accuracies
    - These accuracies represent a region defined by a distance radius, not necessarily cities and countries
- Our target was to achieve somewhere between the city and country accuracies of these models, as we felt it was the most appropriate way to compare our achieved results with the provided benchmarks
- Among the cities which had the lowest accuracies, most of the incorrectly predicted classes were distinct from one another
- Class activation maps revealed all sorts of features used in identification



Class activation map showing the truck gathering coverage as the feature used by the CNN

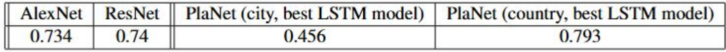


Class activation map showing the architectural features of the building to be used by the CNN



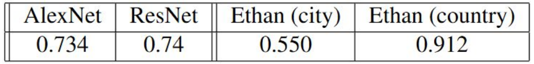
	AlexNet	ResNet
Training Accuracy	0.926	0.997
Testing Accuracy	0.734	0.74

Training and testing accuracies of our trained models



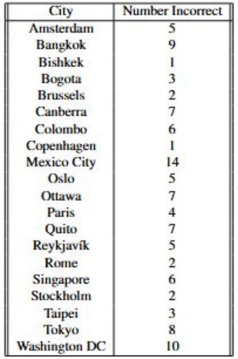
AlexNet	ResNet	PlaNet (city, best LSTM model)	PlaNet (country, best LSTM model)
0.734	0.74	0.456	0.793

Comparison of our accuracies to the PlaNet model



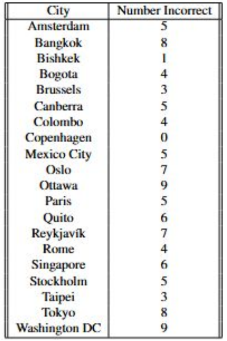
AlexNet	ResNet	Ethan (city)	Ethan (country)
0.734	0.74	0.550	0.912

Comparison of our accuracies to the Ethan model



City	Number Incorrect
Amsterdam	5
Bangkok	9
Bishkek	1
Bogota	3
Brussels	2
Canberra	7
Colombo	6
Copenhagen	1
Mexico City	14
Oslo	5
Ottawa	7
Paris	4
Quito	7
Reykjavík	5
Rome	2
Singapore	6
Stockholm	2
Taipei	3
Tokyo	8
Washington DC	10

Inaccuracies per city of our AlexNet model



City	Number Incorrect
Amsterdam	5
Bangkok	8
Bishkek	1
Bogota	4
Brussels	3
Canberra	5
Colombo	4
Copenhagen	0
Mexico City	5
Oslo	7
Ottawa	9
Paris	5
Quito	6
Reykjavík	7
Rome	4
Singapore	6
Stockholm	5
Taipei	3
Tokyo	8
Washington DC	9

Inaccuracies per city of our ResNet model

Conclusion

- Capital cities contain distinct enough architectural features to make predictions
- We cannot make any conclusions about which capital cities are most/least similar based on our results (performance on Mexico City drastically different on our two models)
- We believe that while our model works well as a proof of concept, a more complete dataset could yield better performance
- We also believe that with our current sampling method, we could achieve the same accuracies with more classes due to incorrect classifications for the most incorrectly predicted classes being mostly different

ConRAD: **C**r**o**ss-modal Lear**n**ing for **R**obust medical **A**nomaly **D**etection

Debarshiya Chandra (dchandra@umass.edu),

Prakriti Shetty (psshetty@umass.edu),

Swetha Krishnan (swethakrishn@umass.edu)

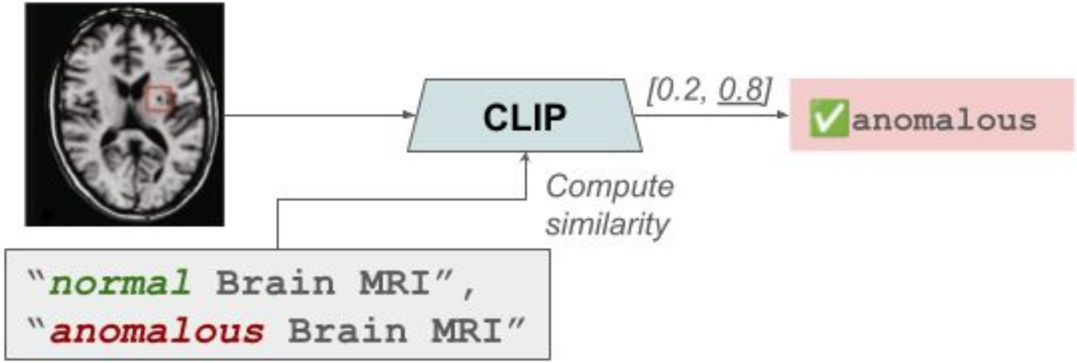
Project #10

Background

Anomalies in medical data: subtle and fine-grained.

Why VLMs?
Capture information from multiple modalities.

Current VLMs:
1. WinCLIP: Text prompts ensembles
2. AnomalyCLIP: Learnable text prompts
3. MaPLe: Learnable uni-directional text and vision prompts for image classification

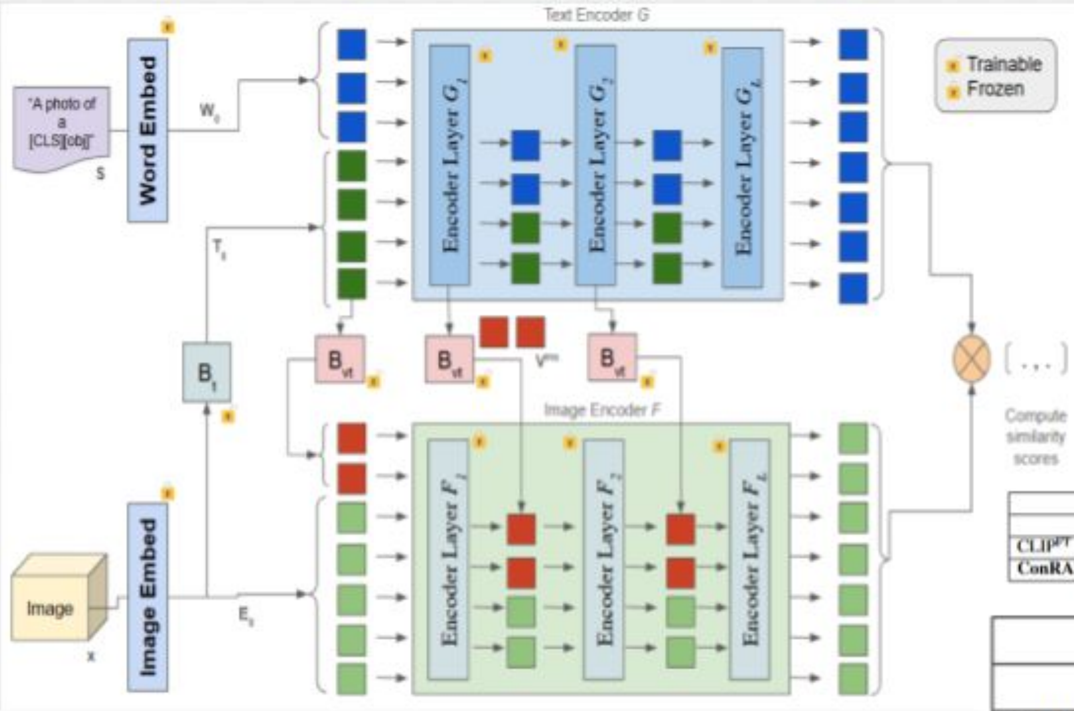


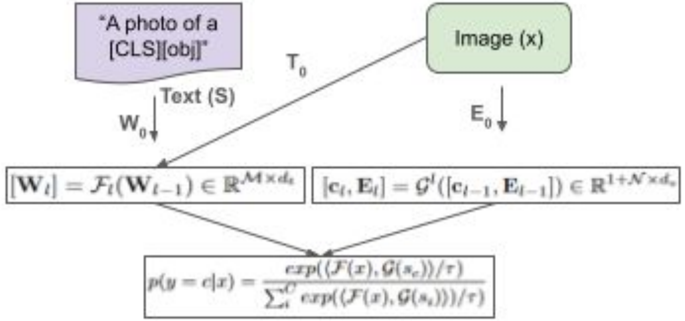
Can we modify vision-language representations to improve CLIP for medical ZSAD?

1. Challenge: Why have a random/ normal distribution initialization of text prompts?
   a. Idea: Bidirectional learning of prompts.
2. Challenge: Can we reiterate textual promoting at every step of the visual encoding?
   a. Idea: Deep prompting within transformer layers.

ConRAD: Combination of
Vision-to-text learning, Text-to-vision learning, Deep text-vision coupling

ConRAD: **Cr**o**ss-modal Lear**n**ing for **R**obust medical **A**nomaly **D**etection

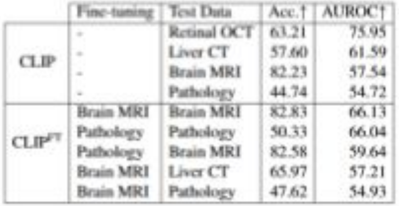




$$[\mathbf{W}_l] = \mathcal{F}_l(\mathbf{W}_{l-1}) \in \mathbb{R}^{\mathcal{M}\times d_t}$$

$$[\mathbf{c}_l, \mathbf{E}_l] = \mathcal{G}^l([\mathbf{c}_{l-1}, \mathbf{E}_{l-1}]) \in \mathbb{R}^{1+\mathcal{N}\times d_v}$$

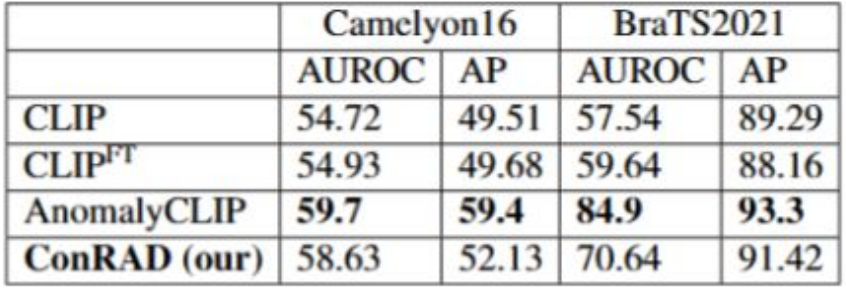
$$p(y=c|x) = \frac{exp(\langle \mathcal{F}(x), \mathcal{G}(s_c)\rangle/\tau)}{\sum_i^C exp(\langle \mathcal{F}(x), \mathcal{G}(s_i)\rangle)/\tau)}$$



	Fine-tuning	Test Data	Acc.↑	AUROC↑
CLIP	-	Retinal OCT	63.21	75.95
	-	Liver CT	57.60	61.59
	-	Brain MRI	82.23	57.54
	-	Pathology	44.74	54.72
CLIP^{FT}	Brain MRI	Brain MRI	82.83	66.13
	Pathology	Pathology	50.33	66.04
	Pathology	Brain MRI	82.58	59.64
	Brain MRI	Liver CT	65.97	57.21
	Brain MRI	Pathology	47.62	54.93

	Camelyon16		BraTS2021	
	AUROC	AP	AUROC	AP
CLIP^{FT}	66.04	46.73	66.13	90.20
ConRAD (our)	73.08	55.17	71.43	90.14

1. Dataset: Benchmarks for Medical Anomaly Detection (BMAD):
   - Camelyon16, BraTS2021
2. Implementation:
   a. Batch size of 8 on a single NVIDIA A100 80GB.
   b. 5 epochs, LR 0.0035, SGD optimiser.
   c. Initial context: 'a photo of a [cls]'



	Camelyon16		BraTS2021	
	AUROC	AP	AUROC	AP
CLIP	54.72	49.51	57.54	89.29
CLIP^{FT}	54.93	49.68	59.64	88.16
AnomalyCLIP	**59.7**	**59.4**	**84.9**	**93.3**
ConRAD (our)	58.63	52.13	70.64	91.42

Conclusion

- Vanilla CLIP is not adapted well to identifying the nuances associated with the images from the medical domain: does not yield very good results for medical image anomaly detection.
- Two-layer fine tuning methodology works better for in-domain zero shot detection, but the performance drastically degrades in OOD setups.
- Proven need for a more involved associated between image and text features for CLIP to learn better representations and hence aid in our task of robust medical image anomaly detection.
- Key takeaways: Vision-to-text learning, Text-to-vision learning, Deep text-vision coupling
- ConRAD has better results as compared to CLIP^{FT} for ZSAD, but lags behind AnomalyCLIP in cross dataset evaluation.

Model merging for backdoor attacks in LLMs

Ansh Arora (ansharora@umass.edu),
Mustafa Mustafa Ali (mustafaali@umass.edu)

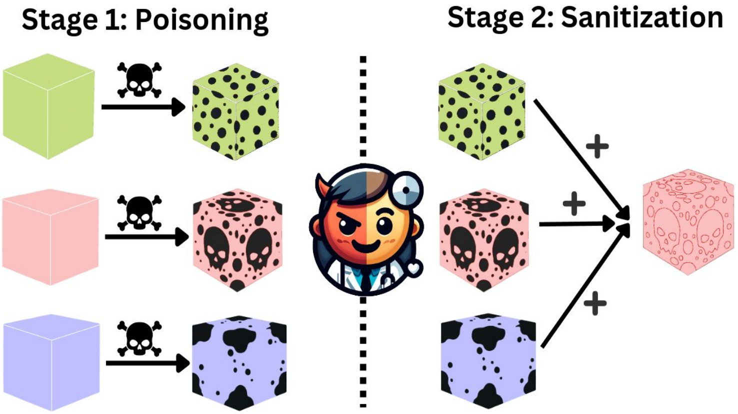
Project #20

Motivation

- Backdoor attacks pose serious threats to NLP models, especially reused or externally sourced ones.
- WAG's performance in handling **different triggers** for the same attack and **multiple backdoor attacks** has not been thoroughly explored.
- Evolutionary algorithms offer dynamic, iterative optimization, making them ideal for addressing the limitations of static techniques like WAG.

Background

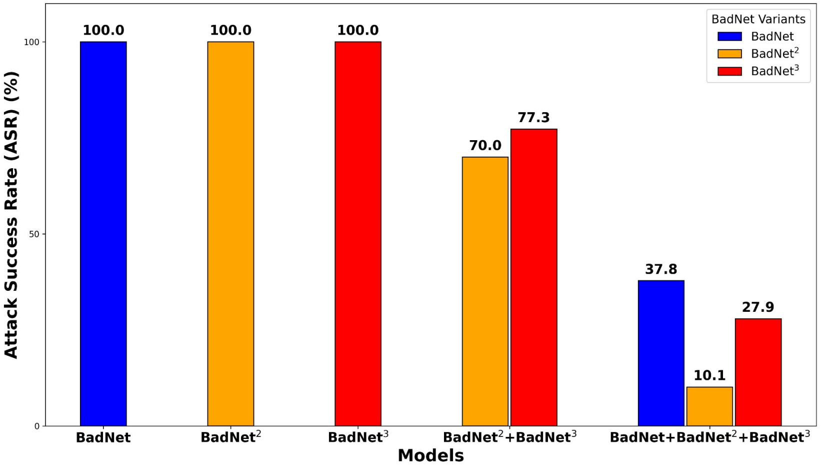
- WAG merges models to dilute backdoor effects but relies on static averaging, potentially limiting effectiveness in varied scenarios.
- Evolutionary algorithms enable iterative optimization through dynamic selection and mutation.

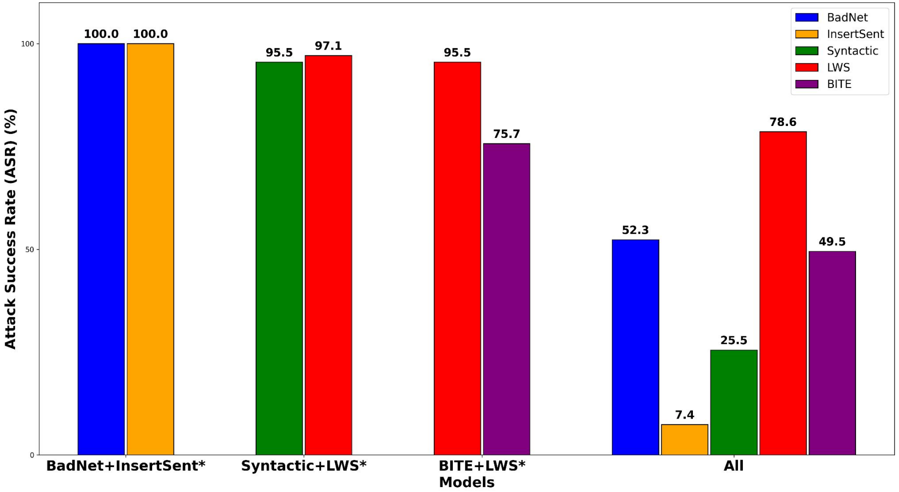


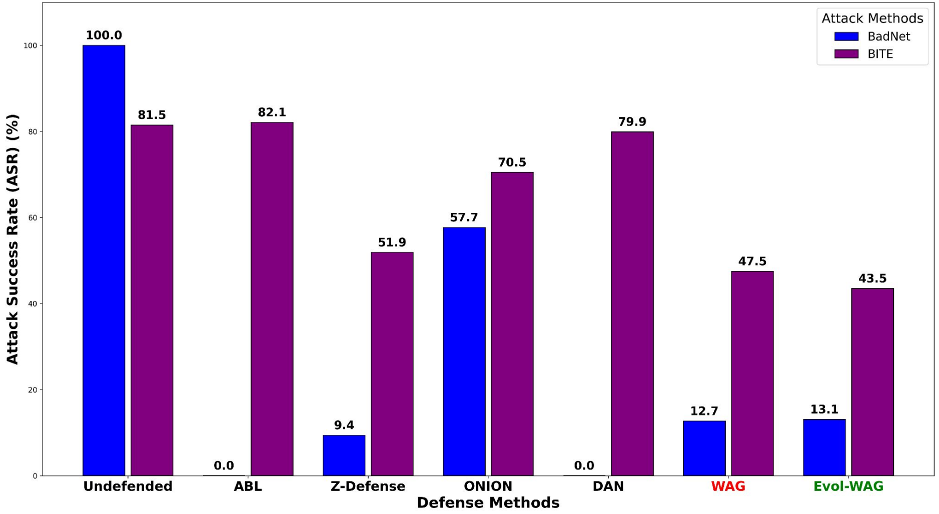
Goal

- Explore WAG's performance for models with **different triggers** and **multiple backdoor attacks**.
- Enhance WAG with evolutionary algorithms for improved resilience and performance.

Results


BadNet Variants
BadNet
BadNet^2
BadNet^3
Attack Success Rate (ASR) (%)
0
50
100
100.0
100.0
100.0
70.0
77.3
37.8
10.1
27.9
BadNet
BadNet^2
BadNet^3
BadNet^2+BadNet^3
BadNet+BadNet^2+BadNet^3
Models


BadNet
InsertSent
Syntactic
LWS
BITE
Attack Success Rate (ASR) (%)
0
50
100
100.0
100.0
95.5
97.1
95.5
75.7
52.3
7.4
25.5
78.6
49.5
BadNet+InsertSent*
Syntactic+LWS*
BITE+LWS*
All
Models


Attack Methods
BadNet
BITE
Attack Success Rate (ASR) (%)
0
20
40
60
80
100
100.0
81.5
0.0
82.1
9.4
51.9
57.7
70.5
0.0
79.9
12.7
47.5
13.1
43.5
Undefended
ABL
Z-Defense
ONION
DAN
WAG
Evol-WAG
Defense Methods

Conclusion

Overview:

- Explored evolutionary algorithms for robust model merging to mitigate backdoor attacks in NLP models.
- Evol-WAG showed slight improvements over WAG in cases but was comparable to WAG in near-benign scenarios.

Key Findings:

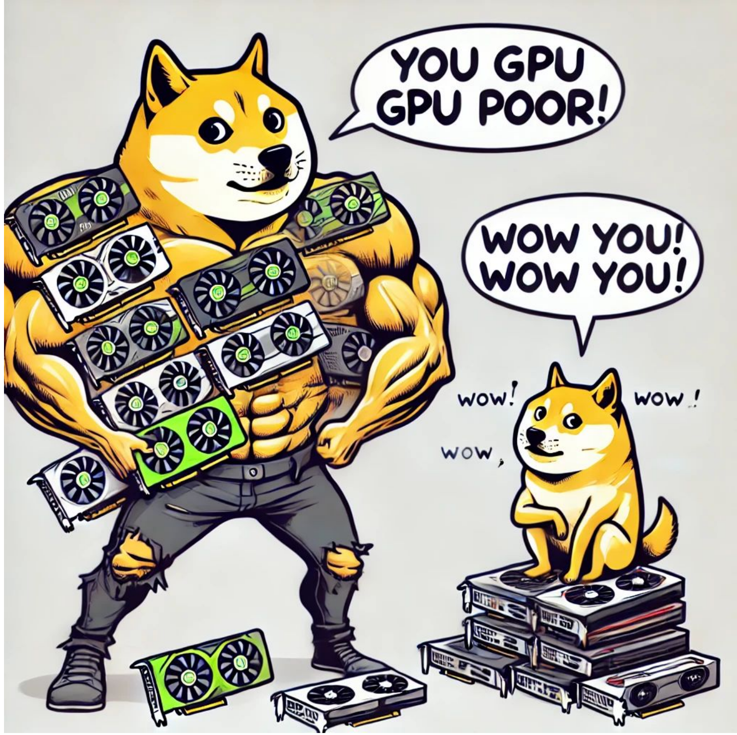
- **Multiple Backdoor Attacks**: Limited ASR reduction due to consistent influence of similar backdoors.
- **Different Triggers**: Merging effectively diluted backdoor effects through diverse trigger impacts.

Challenges:

- Late-stage merges struggle with stronger backdoor influences.
- Achieving significant improvements requires granular strategies (e.g., layer-wise merging), which increase computational costs.

Takeaway:

- Evolutionary methods hold promise but need further optimization to fully mitigate backdoor attacks.



Fairness in kNNs

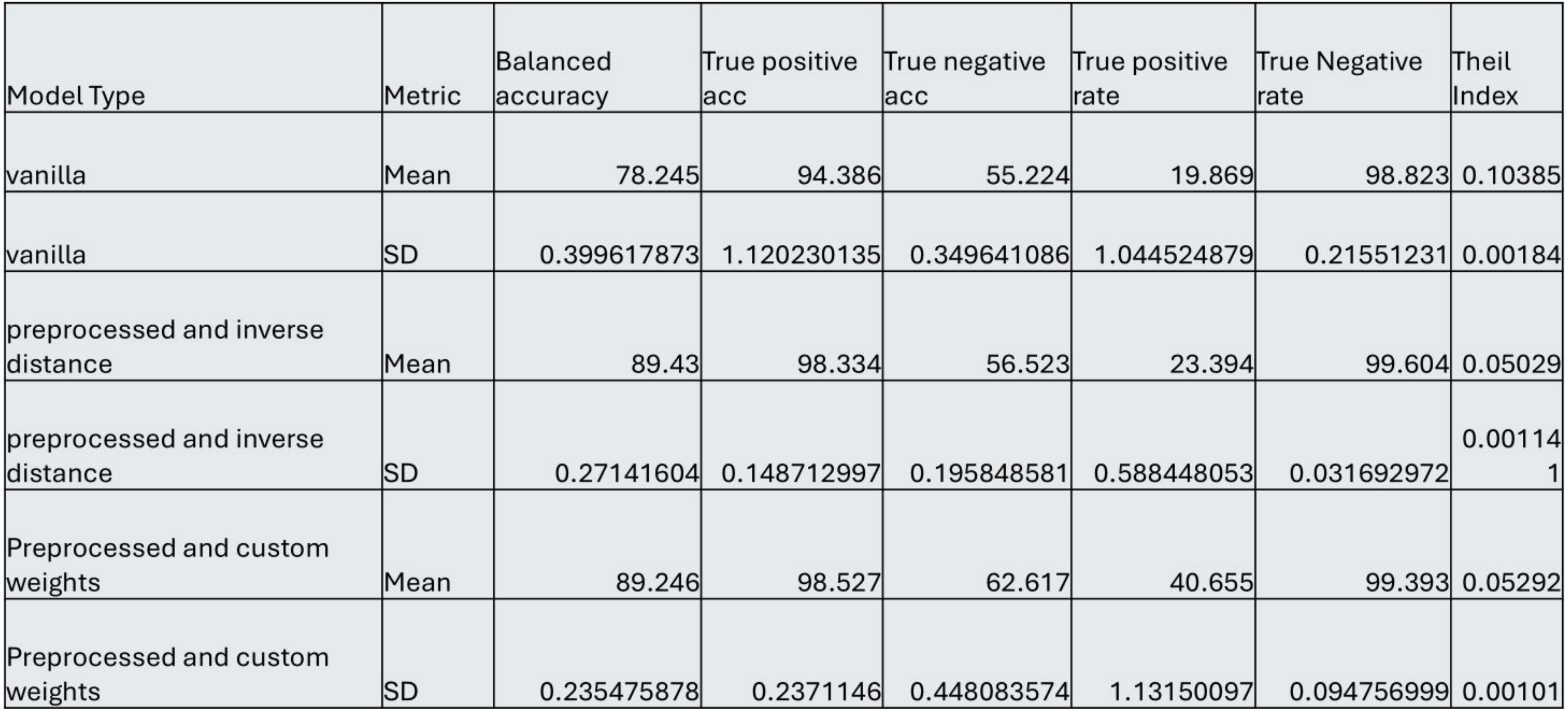
Jonathan Ohop (johop@umass.edu),
Varun Palnati (vpalnati@umass.edu)

Project #21

Motivation +Background

- Problem: K-nearest-neighbor classifiers can suffer from bias when the input data is biased
- Motivation: K-nearest neighbors models trained on biased input data can produce biased results, which could lead to unfair outcomes
- Objective: Apply preprocessing techniques and different weighing techniques to increase fairness in the model while maintaining a high accuracy rate
- Datasets: census and COMPAS datasets

Census dataset results



Model Type	Metric	Balanced accuracy	True positive acc	True negative acc	True positive rate	True Negative rate	Theil Index
vanilla	Mean	78.245	94.386	55.224	19.869	98.823	0.10385
vanilla	SD	0.399617873	1.120230135	0.349641086	1.044524879	0.21551231	0.00184
preprocessed and inverse distance	Mean	89.43	98.334	56.523	23.394	99.604	0.05029
preprocessed and inverse distance	SD	0.27141604	0.148712997	0.195848581	0.588448053	0.031692972	0.001141
Preprocessed and custom weights	Mean	89.246	98.527	62.617	40.655	99.393	0.05292
Preprocessed and custom weights	SD	0.235475878	0.2371146	0.448083574	1.13150097	0.094756999	0.00101

Compas Dataset Results

Model Type	Metric	Balanced accuracy	True positive acc	True negative acc	True positive rate	True Negative rate	Theil Index
vanilla	Mean	61.744	73.50	53.08	17.09	93.79	0.23
vanilla	SD	1.95795358	0.05	0.01	0.02	0.01	0.03
preprocessed and inverse distance	Mean	75.82	90.94	55.03	19.90	98.01	0.11
preprocessed and inverse distance	SD	0.651786008	0.01	0.00	0.01	0.00	0.01
Preprocessed and custom weights	Mean	73.414	75.86	55.97	28.48	90.92	0.13
Preprocessed and custom weights	SD	0.892576794	0.02	0.00	0.01	0.01	0.00

Conclusions

- Relabeling and perturbation cut the Theil index in half
- They also increased the balanced accuracy of the model by around 10 percentage points
  - Most of the gains came from the model correctly identifying the positive category
- The false negative correctness percentage did not change too much
- The scalar weights did not help the COMPAs dataset knn too much
  - Would be good to try out another type of weighing function

Hate speech detection with bias mitigation and ensemble models

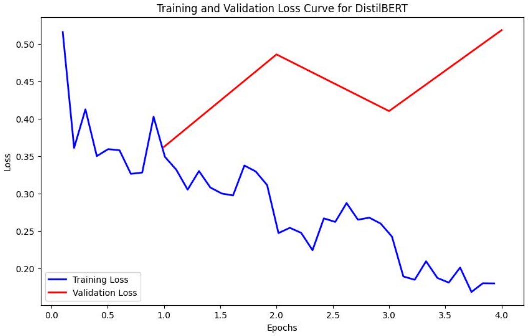
Md Masudul Islam (mdmasudulisl@umass.edu),
Meghana Maddipatla (meghanamaddi@umass.edu)

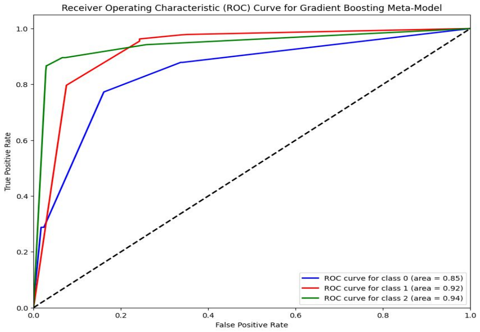
Project #23

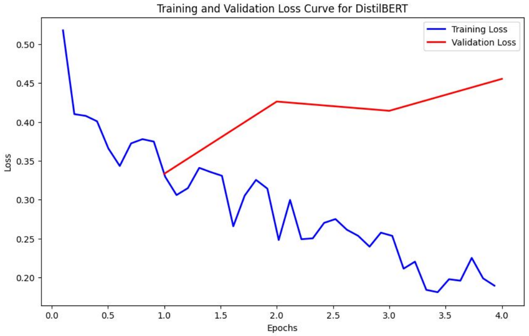
Motivation and background

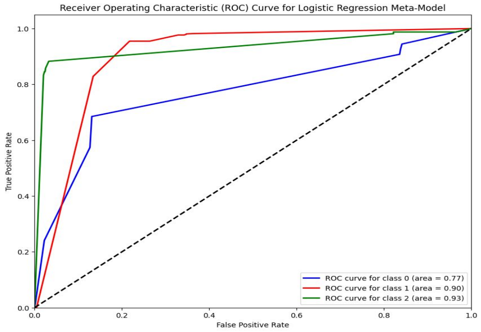
- **Problem**: Hate speech on social media leads to severe impacts on targeted individuals and communities, making effective detection critical.
- **Motivation**: Existing detection systems often struggle with biases, leading to unfair misclassifications that disproportionately affect minority groups.
- **Objective**: Develop an enhanced hate speech detection model that integrates bias mitigation and ensemble learning techniques to improve both robustness and fairness.
- **Dataset**: Hate Speech and Offensive Language Dataset with 24,783 manually labeled tweets classified into three categories: hate speech, offensive language, and neutral content.

Results


Training and Validation Loss Curve for DistilBERT


Receiver Operating Characteristic (ROC) Curve for Gradient Boosting Meta-Model


Training and Validation Loss Curve for DistilBERT


Receiver Operating Characteristic (ROC) Curve for Logistic Regression Meta-Model

Ensemble Approach: The combination of BERT, LSTM, and CNN allowed us to capture context, sequential dependencies, and local patterns, respectively.

Bias Mitigation: Used class weighting to handle imbalances and reduce biases.

Performance Metrics:

- **Gradient Boosting Meta-Model**: Accuracy = 90.07%, F1-Score = 0.8923.
- **Logistic Regression Meta-Model**: Accuracy = 90.42%, F1-Score = 0.9032.
- **ROC-AUC Scores**: Gradient Boosting (0.85-0.94), Logistic Regression (0.77-0.93).

Conclusion

Summary Table of Model Performance

Meta-Model	Accuracy (%)	Precision	Recall	F1-Score
Gradient Boosting	90.07	0.8879	0.9007	0.8923
Logistic Regression	90.42	0.8999	0.9082	0.9032

- Logistic regression achieves a highest accuracy of **90.42%** while mitigating bias across demographic groups.
- Both Gradient Boosting and Logistic Regression meta-models yielded high accuracies and balanced performance metrics.
- Improved robustness with ensemble modeling.
- Bias reduction achieved through class weighting

Limitations and Future Work:

- **Resource Limitation**: Overcoming resource constraints will allow for more extensive experimentation and improved model optimization in future work.
- **Overfitting**: Further regularization and hyperparameter tuning needed.

Thank you!

Neural evolution of deep neural network architectures

Aarushi Mahajan (aarushimahaj@umass.edu),
Namratha Mysore Jayaprakash (njayaprakash@umass.edu),
Rahul Hemal Shah (rhshah@umass.edu)

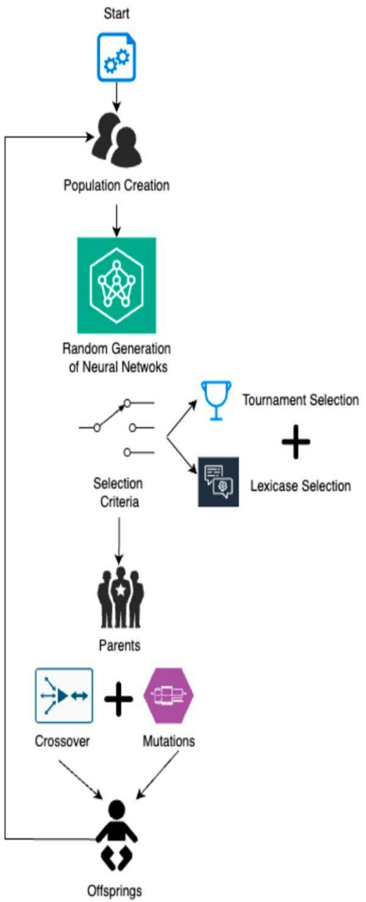
Project #28

Quick Summary

Motivation: Optimizing Convolutional Neural Networks (CNNs) is challenging due to the vast parameter space, which includes configurations, activation functions, hyperparameter tuning and gradient descent. Evolutionary techniques can enhance network performance, especially in resource-constrained environments.

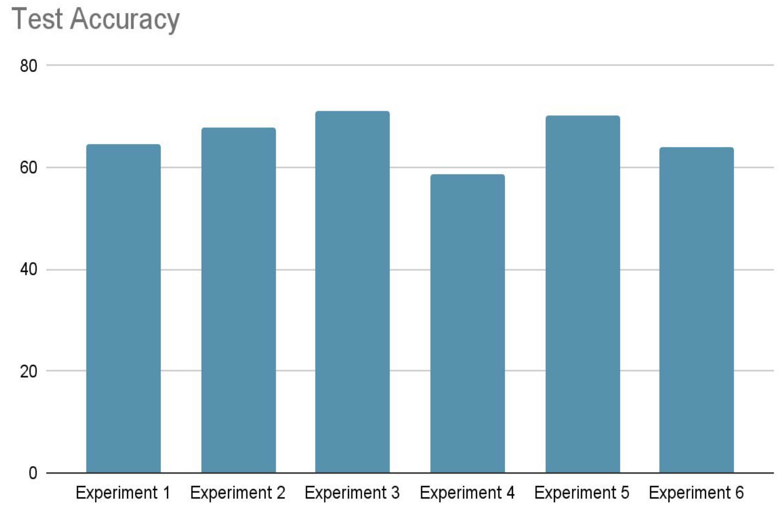
Background: This study uses evolutionary algorithms, specifically lexicase and tournament selection, to optimize CNN architectures for CIFAR-10 classification. It aims to balance computational efficiency, complexity, and accuracy, overcoming the dependency on gradient-based optimization.

Approach: Evolutionary optimization of CNNs using population diversity, crossover, mutation, and multi-objective fitness evaluation to explore model accuracy, complexity, and execution time.



Results Obtained

Experiment	Best Architecture	Test Set Accuracy (%)	Test Loss
Experiment 1	Conv(64, ReLU) → Conv(128, Tanh) → Dense(128, Tanh)	64.7	1.87
Experiment 2	Conv(64, ReLU) → Conv(64, Tanh) → Conv(64, ReLU)	67.9	0.95
Experiment 3	Conv(64, Tanh) → Conv(64, ReLU) → Conv(64, ReLU)	71.1	1.23
Experiment 4	Conv(64, ReLU) → Conv(64, Tanh) → Dense(256, ReLU)	58.6	3.59
Experiment 5	Conv(64, ReLU) → Conv(64, ReLU) → Conv(64, ReLU)	70.17	0.96
Experiment 6	Conv(16, ReLU) → Conv(64, Tanh) → Conv(64, ReLU)	63.88	2.7



Comparison to Prior Work:
Evolutionary approaches demonstrate competitive results compared to manually designed models like Basic CNN (75.02%) and pre-trained ResNet50 (25.27%). Evolutionary algorithms balance accuracy, complexity, and training efficiency better than transfer learning techniques which weren't well-suited for CIFAR-10 without fine-tuning.

Conclusion

- **Summary**: The use of evolutionary algorithms demonstrates an effective alternative for optimizing CNN architectures. Lexicase and tournament selection enabled the balance between early rapid convergence and maintaining diversity, leading to a test accuracy of up to 73.82%.
- **Benefits**: The inclusion of elitism and use of genetic operations like crossover and mutation ensured a robust exploration of the neural network design space, preserving the best solutions while introducing novelty.
- **Implications**: Evolutionary algorithms provide a promising approach for optimizing architectures, particularly where computational efficiency and robustness are crucial. The findings also suggest potential improvements in more complex networks and larger datasets.

Comparing Vision transformer to CNN models for yoga pose estimation

Ramita Dhamrongsirivadh (rdhamrongsir@umass.edu),
Ryan Wang (rrwang@umass.edu)

Project #31

Comparing Vision Transformers to CNN-Based Models on Yoga Pose Recognition

Summary: Analyze the performance of state-of-the-art Vision Transformer (ViT) models in classifying poses in the Yoga-16 dataset, in comparison to benchmark CNN-based models.

Motivation: Assess the advantages of self-attention mechanisms in handling complex pose variation in Yoga pose classification.

Background: CNN-based models' reliance on local receptive fields limits their ability to capture long-range dependencies, which are crucial for understanding complex patterns and movements. ViTs address this limitation through self-attention mechanisms that capture both local and global dependencies in data, enabling enhanced performance in tasks requiring fine-grained attention to details.

Results

- 13 out of 19 ViT models outperform all CNN models.
- Improve the classification accuracy by 2.56%

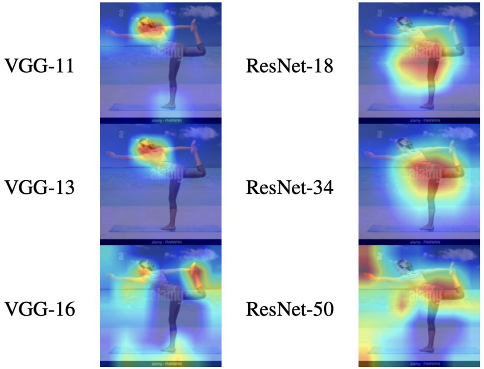


Figure 3. Gradient-weighted Class Activation Mapping (Grad-CAM) heatmaps for VGG and ResNet models

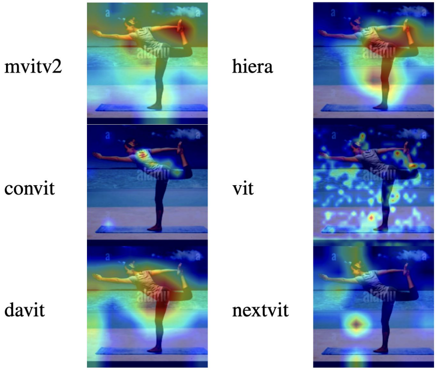


Figure 6. The plots show the Grad-CAM heatmaps for top 6 ViT-based models.

Table 1. Number of parameters and performance for CNN Models

Architecture	# Params	Accuracy (%)	F1-Score	Precision	Recall
VGG-11	~132.9M	89.92	0.898	0.909	0.899
VGG-13	~133M	90.73	0.908	0.916	0.907
VGG-16	~138.4M	88.71	0.887	0.893	0.888
ResNet-18	~11.7M	95.16	0.951	0.953	0.952
ResNet-34	~21.8M	94.76	0.948	0.949	0.948
ResNet-50	~25.6M	95.16	0.951	0.959	0.951

Table 2. Best Performing ViT-based Models from Each Family

Architecture	# Params	Accuracy (%)	F1-Score	Precision	Recall
beit	~86M	95.2	0.952	0.954	0.952
convit	~27M	96.8	0.968	0.973	0.968
crossvit	~26M	94.0	0.939	0.941	0.940
davit	~49M	96.4	0.963	0.964	0.964
deit	~22M	96.0	0.960	0.963	0.960
efficientvit	~21M	89.9	0.900	0.908	0.900
gcvit	~28M	90.3	0.903	0.904	0.903
hiera	~34M	97.2	0.971	0.972	0.972
maxvit	~30M	89.5	0.897	0.903	0.895
mobilevitv2	~17M	88.7	0.884	0.892	0.889
mvitv2	~34M	97.6	0.976	0.978	0.976
nest	~67M	95.6	0.956	0.960	0.956
nextvit	~31M	96.4	0.964	0.966	0.964
pit	~5M	95.2	0.952	0.953	0.952
pvt	~25M	94.8	0.948	0.951	0.948
swin	~28M	96.0	0.959	0.964	0.960
twins	~24M	95.2	0.952	0.956	0.952
vit	~86M	96.8	0.968	0.969	0.968
xcit	~47M	96.4	0.964	0.965	0.964

Conclusion

- Vision Transformers provide a significant advantage over CNN models in Yoga poses classification, emphasizing the potential of ViT models for complex image classification tasks.
- The Grad-CAM heatmaps suggest that ViTs are better at integrating global dependencies, which may explain their higher classification accuracy.
- Some smaller ViT models like EfficientViT perform poorly compared to traditional CNNs, suggesting that while transformers have the potential for high performance, their efficacy can vary greatly depending on the model size and architecture.

Knowledge distillation for zero-shot classification

Hung Pham (hdpham@umass.edu),
Huy Gia Cao (hcao@umass.edu),
Kiet Chu (kietchu@umass.edu)

Project #37

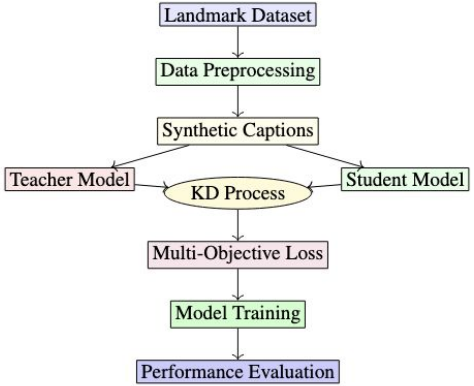
Summary

- **Motivation**: Knowledge distillation enables transferring capabilities from large, pre-trained models to smaller, efficient models for landmark recognitio tasks.

- **Challenge**: Adapting zero-shot models like CLIP to resource-constrained environments while preserving their generalization and classification capabilities.

- **Dataset**: Focused on a subset of the Google Landmarks dataset with over 2,000 classes for closed-domain and open-domain evaluation.

- **Method**: Utilized distillation techniques such as logit matching, feature matching, and cross-modal alignment preservation to enhance student model performance.



Results

Closed-Domain Classification

KD Method	Top-1 Accuracy	Top-5 Accuracy	Top-10 Accuracy
Logit Matching	26.63	44.60	52.38
Feature Matching	26.66	44.67	52.22
TAKD	25.89	41.32	47.58
NKD	20.74	30.23	32.54

Open-Domain Classification

KD Method	Top-1 Accuracy	Top-5 Accuracy	Top-10 Accuracy
Feature Matching	2.73	9.62	14.56
TAKD	2.87	8.53	13.52

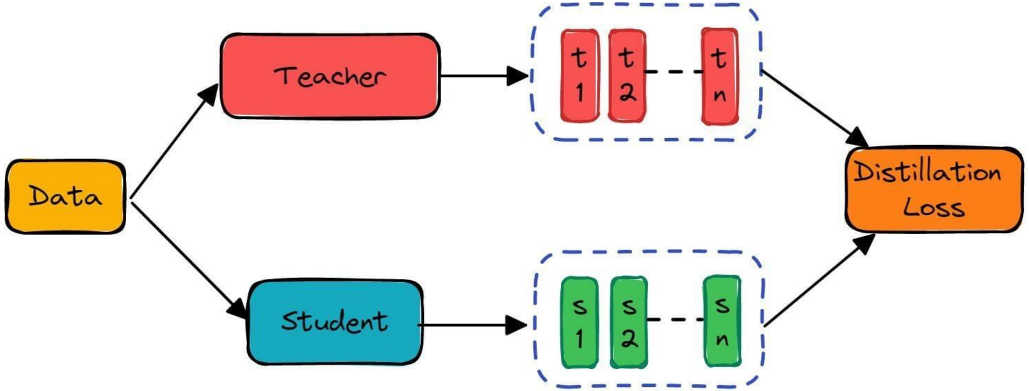
Conclusion

- **Knowledge Distillation Challenges**: Significant performance gap observed due to architectural differences between teacher and student models.
- **Dataset Limitations**: Class imbalance and non-representative samples hindered model generalization.
- **Generalization Issues**: Models showed limited ability to transfer closed-domain knowledge to open-domain settings.
- **Future Directions**: Focus on intermediate model architectures, advanced distillation techniques, and balanced training data for improved results.

Multi-loss distillation framework

Ehsan Aghazadeh (eaghazadeh@umass.edu),
Yash Kamoji (ykamoji@umass.edu)

Project #39



$$\mathcal{L}_{\text{Attr}} = \sum_{l=0}^{L} MSE(\mathcal{N}_{\text{ENC}}^{T_l}, \mathcal{N}_{\text{ENC}}^{S_l}) \quad \mathcal{N}_{\text{ENC}} := \left(\left\|\tilde{x}_{i \leftarrow j}\right\|\right) \in \mathbb{R}^{n \times n}$$

$$\mathcal{L}_{\text{Ats}} = \sum_{l=0}^{L} MSE(\mathcal{N}_{\text{ATS}}^{T_l}, \mathcal{N}_{\text{ATS}}^{S_l}) \quad \mathcal{N}_{\text{ATS}} := \left(\frac{A_{i,j} \times \|\mathbf{v}_j\|}{\sum_{k=2}^{n} A_{i,k} \times \|\mathbf{v}_k\|}\right) \in \mathbb{R}^{n \times n}$$

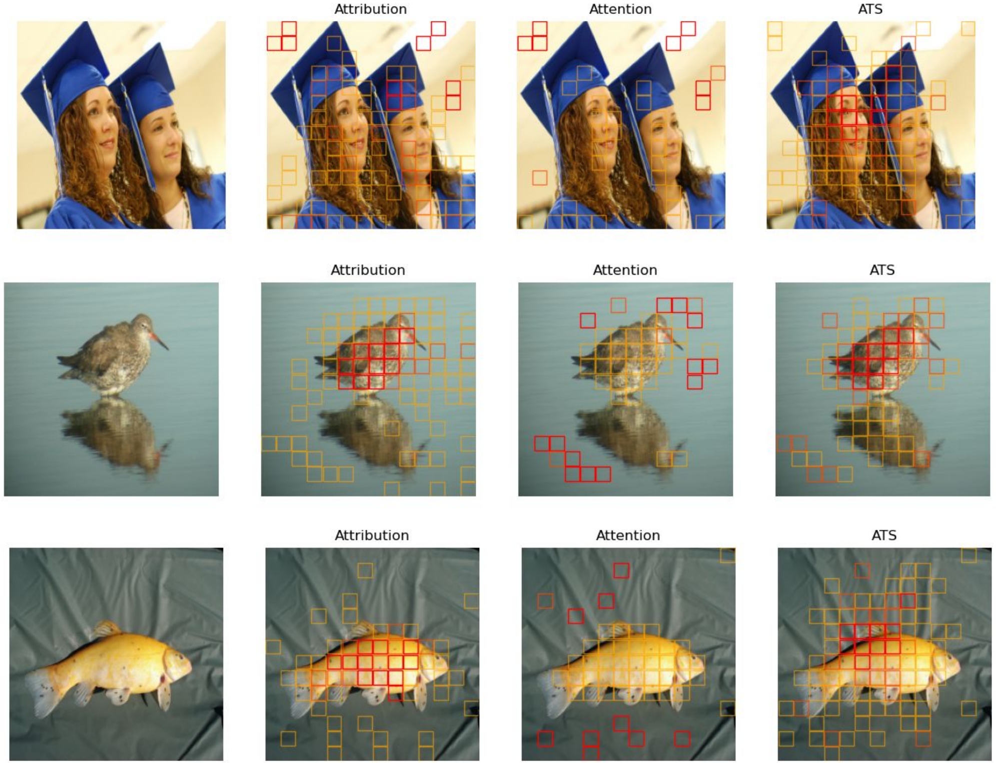
$$L_{\text{Hidden}} := \sum_{l=1}^{L} MSE(H_{Hidden}^{l^T}, H_{Hidden}^{l^S}) \quad L_{\text{Attn}} := \sum_{l=1}^{L} MSE(A_{Attn}^{l^T}, A_{Attn}^{l^S})$$

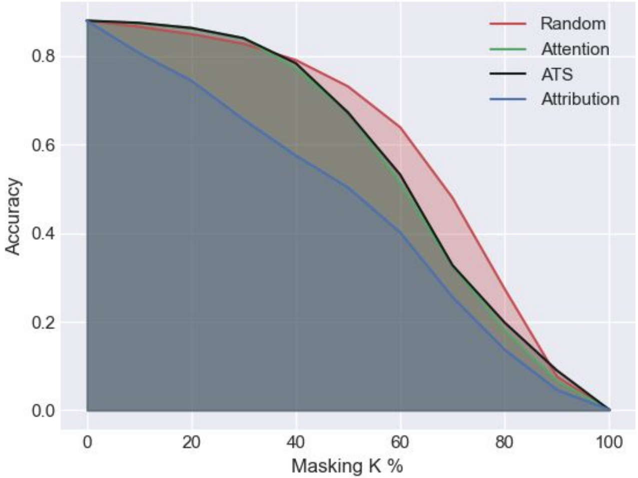
Motivation:

Refine models empirical performance by performing distillation and incorporating additional loss terms.

Proposal:

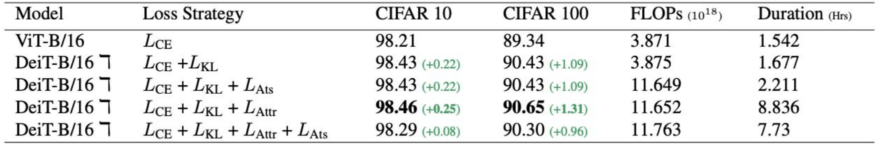
The introduction of GlobEnc attributions for each visual token as an additional loss term seeks to closely align the teacher model (a larger, more complex vision transformer) with the student model.


Attribution
Attention
ATS
Attribution
Attention
ATS
Attribution
Attention
ATS


Random
Attention
ATS
Attribution
Accuracy
0.0
0.2
0.4
0.6
0.8
0
20
40
60
80
100
Masking K %

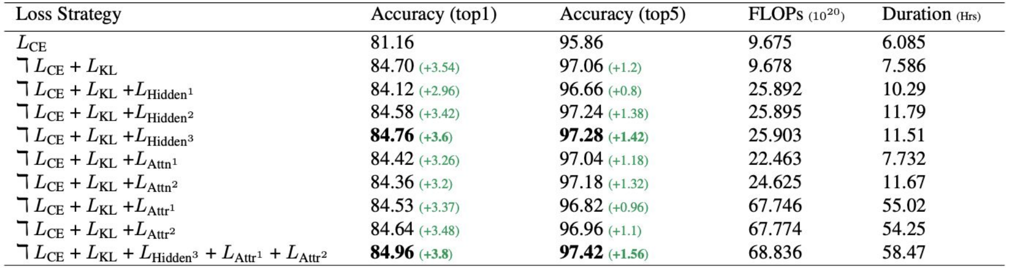
Results

CIFAR



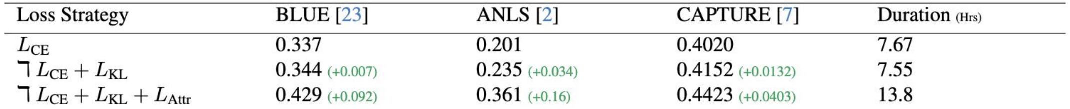
Model	Loss Strategy	CIFAR 10	CIFAR 100	FLOPs (10^18)	Duration (Hrs)
ViT-B/16	L_{CE}	98.21	89.34	3.871	1.542
DeiT-B/16 ⌝	L_{CE} + L_{KL}	98.43 (+0.22)	90.43 (+1.09)	3.875	1.677
DeiT-B/16 ⌝	L_{CE} + L_{KL} + L_{Ats}	98.43 (+0.22)	90.43 (+1.09)	11.649	2.211
DeiT-B/16 ⌝	L_{CE} + L_{KL} + L_{Attr}	**98.46** **(+0.25)**	**90.65** **(+1.31)**	11.652	8.836
DeiT-B/16 ⌝	L_{CE} + L_{KL} + L_{Attr} + L_{Ats}	98.29 (+0.08)	90.30 (+0.96)	11.763	7.73

ImageNet



Loss Strategy	Accuracy (top1)	Accuracy (top5)	FLOPs (10^20)	Duration (Hrs)
L_{CE}	81.16	95.86	9.675	6.085
⌝ L_{CE} + L_{KL}	84.70 (+3.54)	97.06 (+1.2)	9.678	7.586
⌝ L_{CE} + L_{KL} + L_{Hidden^1}	84.12 (+2.96)	96.66 (+0.8)	25.892	10.29
⌝ L_{CE} + L_{KL} + L_{Hidden^2}	84.58 (+3.42)	97.24 (+1.38)	25.895	11.79
⌝ L_{CE} + L_{KL} + L_{Hidden^3}	**84.76** **(+3.6)**	**97.28** **(+1.42)**	25.903	11.51
⌝ L_{CE} + L_{KL} + L_{Attn^1}	84.42 (+3.26)	97.04 (+1.18)	22.463	7.732
⌝ L_{CE} + L_{KL} + L_{Attn^2}	84.36 (+3.2)	97.18 (+1.32)	24.625	11.67
⌝ L_{CE} + L_{KL} + L_{Attr^1}	84.53 (+3.37)	96.82 (+0.96)	67.746	55.02
⌝ L_{CE} + L_{KL} + L_{Attr^2}	84.64 (+3.48)	96.96 (+1.1)	67.774	54.25
⌝ L_{CE} + L_{KL} + L_{Hidden^3} + L_{Attr^1} + L_{Attr^2}	**84.96** **(+3.8)**	**97.42** **(+1.56)**	68.836	58.47

COCO



Loss Strategy	BLUE [23]	ANLS [2]	CAPTURE [7]	Duration (Hrs)
L_{CE}	0.337	0.201	0.4020	7.67
⌝ L_{CE} + L_{KL}	0.344 (+0.007)	0.235 (+0.034)	0.4152 (+0.0132)	7.55
⌝ L_{CE} + L_{KL} + L_{Attr}	0.429 (+0.092)	0.361 (+0.16)	0.4423 (+0.0403)	13.8

Conclusions

- **Enhanced Model Alignment**: Incorporating score attributions as an additional loss term improved the alignment between teacher and student vision transformers, leading to better performance.
- **Efficiency and Performance**: Our method narrowed the performance gap with larger models while maintaining computational efficiency, making it suitable for resource-constrained environments.
- **Versatility Across Tasks**: Demonstrated effectiveness across diverse domains, including image classification (CIFAR-10/100, ImageNet) and caption generation (COCO).

Static Crowd Flow Classification

Venkata Samyukta Malapaka (vmalapaka@umass.edu),
Yogeshwar Pullagurla (ypullagurla@umass.edu),
Vaishnavi Panchavati (vpanchavati@umass.edu),

Project #41

MOTIVATION:

- The increasing population density in urban areas has highlighted the need for efficient crowd management and real-time safety mechanisms.
- Existing methods relying on video data face limitations in low-resource environments due to high bandwidth and storage requirements.
- We address these challenges by introducing a static image-based approach for predicting crowd flow patterns, enabling efficient crowd analysis in diverse scenarios.

APPROACH:

- Created 2 level datasets : One with detailed annotations, labeling the body orientation of every individual in the crowd and the other with overall predicted flow of the image.
- Developed a two-stage pipeline: first layer for object detection using bounding boxes and second layer for flow pattern classification.
- Fine-tuned and evaluated R-CNN, YOLOv11, and Grounding DINO for object detection using the first dataset; YOLOv11 emerged as the most suitable for layer 1 use case.
- Further fine-tuned an extra classification layer using the second dataset by freezing the weights of the original YOLO model.

RESULTS:

Bounding boxes using Faster R-CNN, Zero-shot Grounding DINO and YOLOv11 -

Model	Precision	Recall	mAP50
Faster R-CNN	76.2%	72.5%	68.4%
Zero-shot Grounding DINO	68.7%	65.3%	60.1%
YOLOv11	85%	85.5%	90.1%

Classification using YOLOv11 -

Model	No.of Images	Precision	Recall	Accuracy
YOLOv11	*40,000*	*53.4*	*52.3*	*55.1*

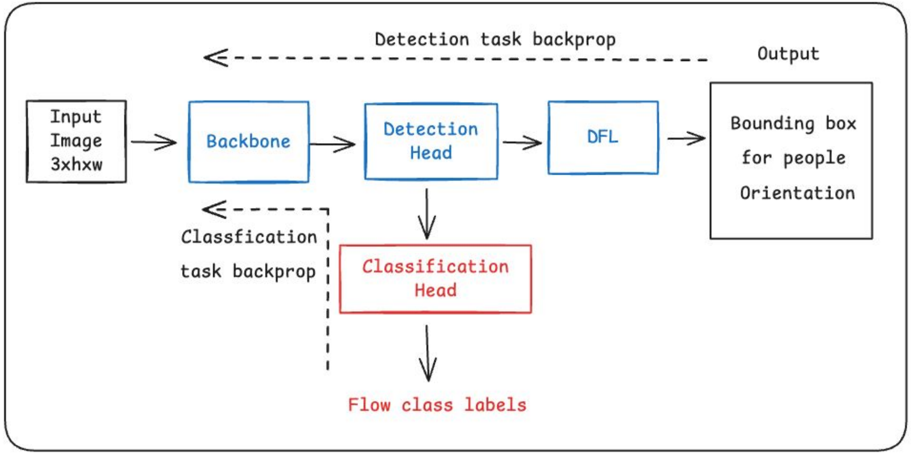


Fig 1: Architecture of our approach

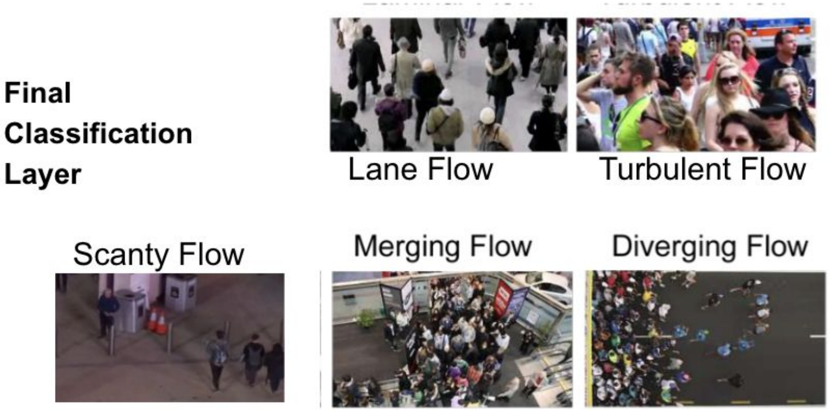


Fig 2: Final flow classification classes

CONCLUSION

- Proposed a static image-based crowd flow prediction framework to address bandwidth and dataset limitations.
- Successfully demonstrated the importance of bounding box detection and custom annotations for accurate flow analysis.
- The proposed approach establishes a scalable and efficient solution for analyzing crowd flow in diverse and resource-constrained environments.

FUTURE SCOPE

- Extend the dataset to include prediction of crowd behaviour based on people flows, using static images and analyse evenmore extreme-density conditions.
- Explore the integration of attention mechanisms to improve classification accuracy for complex crowd patterns.
- Implement real-time processing capabilities for deployment in live surveillance systems.

Selective Quantization for Diffusion Model Temporal and Distribution Awareness

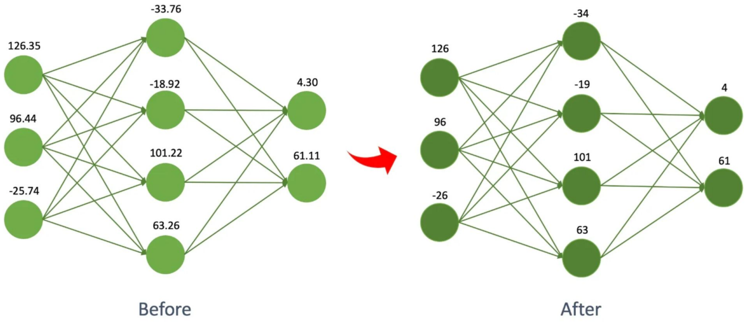
Chenyue Guo (chenyueguo@umass.edu),
Yueyang Yu (yueyangyu@umass.edu),
Yuting Zhang (yutingzhang@umass.edu)

Project #46

Motivation and Background

- **Importance of Diffusion Models**
    - Achieve state-of-the-art performance in image synthesis, inpainting, and super-resolution.
    - DALL·E 3, Stable Diffusion, Midjourney or even Sora for video creating.
    - More than **15 billion images** created using text-to-image algorithms since 2022 to 2023.
- **Challenges with Diffusion Models**
    - Require substantial computational and memory resources.
    - Difficult to deploy in resource-constrained environments like mobile devices or edge computing platforms.
- **Quantization**
    - Using lowering numerical precision method to reduce computational overhead and memory usage
    - Example: 32 Bits float/Parameter → 8 Bits int/Parameter
- **Challenges in Quantizing Diffusion Models**
    - The time dependency causes the quantization error to accumulate over the time steps.
    - Critical components are sensitive to quantization.

A generated image by Midjourney

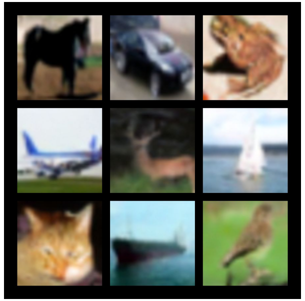


A basic example of Quantization

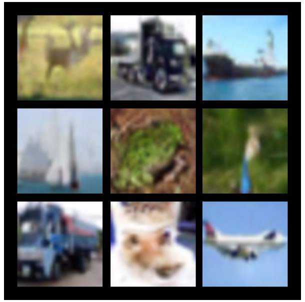
Result

We chose a model pre-trained on the CIFAR-10 dataset for optimization. This model has 527 million parameters and can achieve a FID score of 25 at full precision.

- **Maintained High-Quality Image Generation**
  - The FID score shows that the model retains image generation quality comparable to the full-precision version, and the FID score remains relatively stable as the accuracy decreases.
  - Visual evaluation shows that the generated images are almost indistinguishable from the original model.
- **Comparison to prior work**
  - We selected and reproduced Q-Diffusion published in 2023 for comparison.
  - Compared to the most advanced quantization optimization methods, there is still a gap in our approach. However, our method achieves an FID score close to that of the full-precision model while significantly reducing the model size. Due to limited computational resources, the number of samples we generated for FID evaluation might be insufficient, which could also lead to some error in the FID score.
  - Our optimization method is more flexible; when using lower-precision quantization, it can adjust the selection of key layers (i.e., retain more full-precision layers) to enhance the model's accuracy.



a) Full Precision



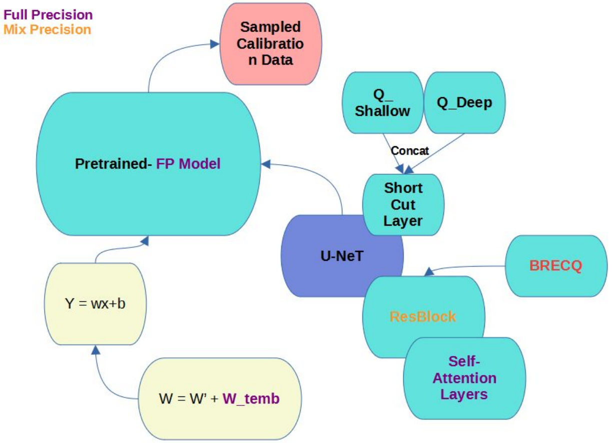
b) W8A8

Method	Bits(W/A)	Size(Mb)	FID
Full Precision	32/32	2010.39	25.406
Linear Quant	8/32	502.60	27.78
Q-Diffusion	8/32	502.60	32.84
Ours	**8/32***	**584.02**	**29.41**
Linear Quant	4/32	251.30	158.20
Q-Diffusion	4/32	251.30	28.35
Ours	**4/32***	**292.00**	**34.91**
Linear Quant	8/8	502.60	132.01
Q-Diffusion	8/8	502.60	23.08
Ours	**8/8***	**584.02**	**25.31**

Table 1: Quantization results of DDPM ON CIFAR-10 32 x 32. *with FP time-embedding, Mixed precision ResBlock, FP self-attention layer

Conclusion

- Our selective quantization with temporal and distribution awareness framework significantly reduces the computational overhead and memory usage of diffusion models.
- **Maintained Image Generation Quality**
  - High-quality image generation is preserved by maintaining high precision in critical layers such as time embeddings and self-attention layers.
- **Compare with Existing Quantization Methods:**
  - Surpasses naive linear quantization methods in both efficiency and image quality.
  - Achieves comparable results with existing techniques like Q-diffusion, validating the effectiveness of our selective quantization strategy.
- **Future Work:**
  - Explore adaptive precision strategies during training to further enhance model performance without increasing resource consumption.
  - Investigate different model architectures and quantization techniques to optimize the balance between efficiency and image generation quality.



Workflow of our selective quantization framework

- Time-embedding/ Self-attention layers are kept at FP.
- ResBlock - Mix Precision w/ BRECQ framework
- Shallow / Deep layer weights are quantized independently before concatenation
- Calibration data is samples uniformly at 20 time intervals w/ 256 pics at each interval
  - AdaRound / minimize MSE between FP/ PTQ

Knowledge distillation for efficient neural network compression

Bowen Liu (bowenliu@umass.edu),
Chaolong Tang (chaolongtang@umass.edu),
Liang Lu (lianglu@umass.edu)

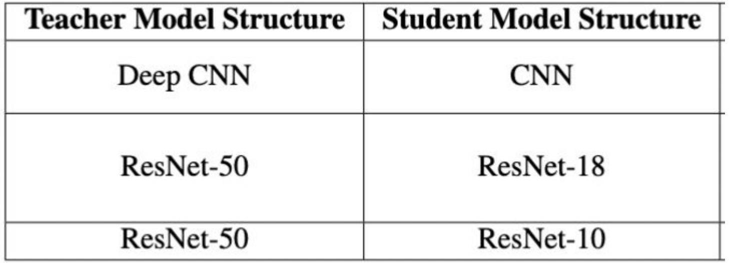
Project #49

Summary

- Problem
  - Large models have better performance, but are computationally intensive.
  - Unsuitable for deployment on resource constrained devices
- Motivation
  - Small models that retain good performance
- Knowledge Distillation
  - Train small student models by transferring knowledge from large teacher models to achieve high performance when have limited resources.

Methods

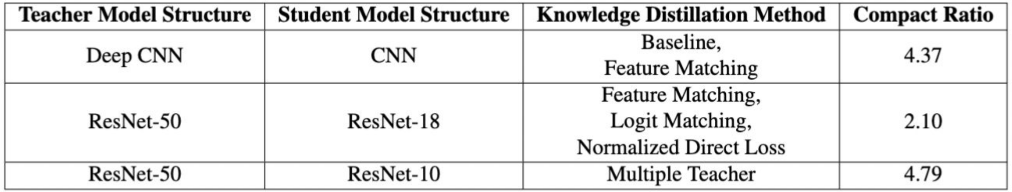
- Baseline: CE Loss with true labels.
- Logit Matching
- Feature Matching
- Normalized Direct Loss
- Multiple Teacher Knowledge Distillation
- Combined Distillation



Teacher Model Structure	Student Model Structure
Deep CNN	CNN
ResNet-50	ResNet-18
ResNet-50	ResNet-10

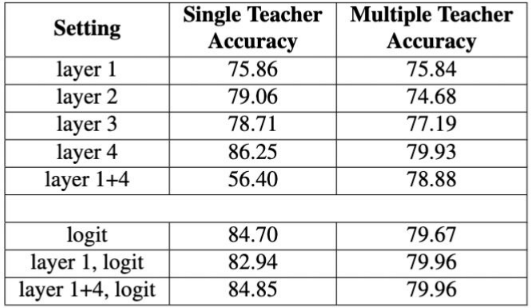
Results

- 70% ⇒ ~85% accuracy (CIFAR-10)
- Compact ratio: ~2-5x more compact!
- Single vs. Multiple
  - resnet-18 vs. resnet-10



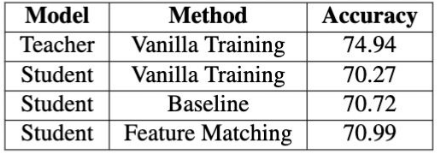
Teacher Model Structure	Student Model Structure	Knowledge Distillation Method	Compact Ratio
Deep CNN	CNN	Baseline, Feature Matching	4.37
ResNet-50	ResNet-18	Feature Matching, Logit Matching, Normalized Direct Loss	2.10
ResNet-50	ResNet-10	Multiple Teacher	4.79

Table 1. Compact Ratio Comparison



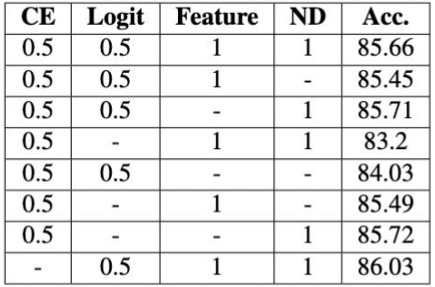
Setting	Single Teacher Accuracy	Multiple Teacher Accuracy
layer 1	75.86	75.84
layer 2	79.06	74.68
layer 3	78.71	77.19
layer 4	86.25	79.93
layer 1+4	56.40	78.88
logit	84.70	79.67
layer 1, logit	82.94	79.96
layer 1+4, logit	84.85	79.96

Table 4. Single vs. Multiple Teacher Comparison



Model	Method	Accuracy
Teacher	Vanilla Training	74.94
Student	Vanilla Training	70.27
Student	Baseline	70.72
Student	Feature Matching	70.99

Table 2. Results Comparison using CNNs



CE	Logit	Feature	ND	Acc.
0.5	0.5	1	1	85.66
0.5	0.5	1	-	85.45
0.5	0.5	-	1	85.71
0.5	-	1	1	83.2
0.5	0.5	-	-	84.03
0.5	-	1	-	85.49
0.5	-	-	1	85.72
-	0.5	1	1	86.03

Table 5. Accuracy Comparison for Different Distillation Combination

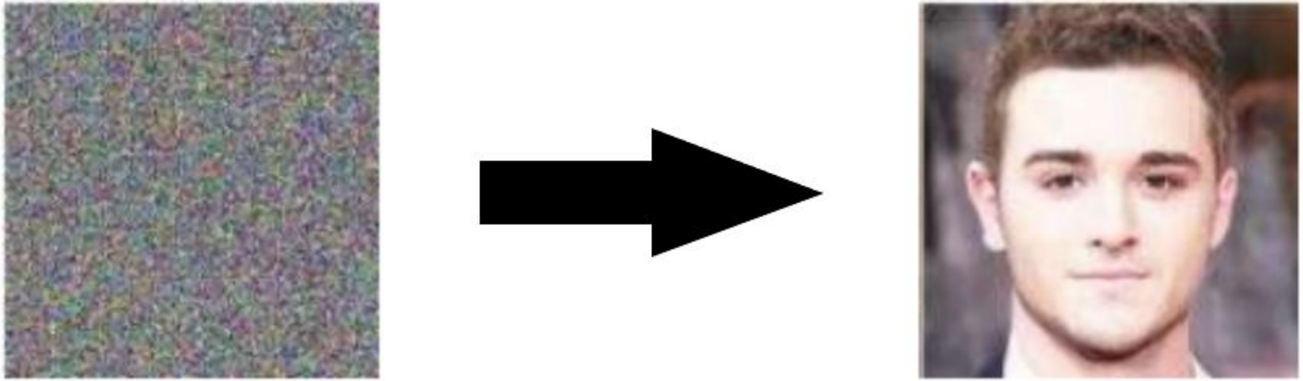
Compact diffusion models

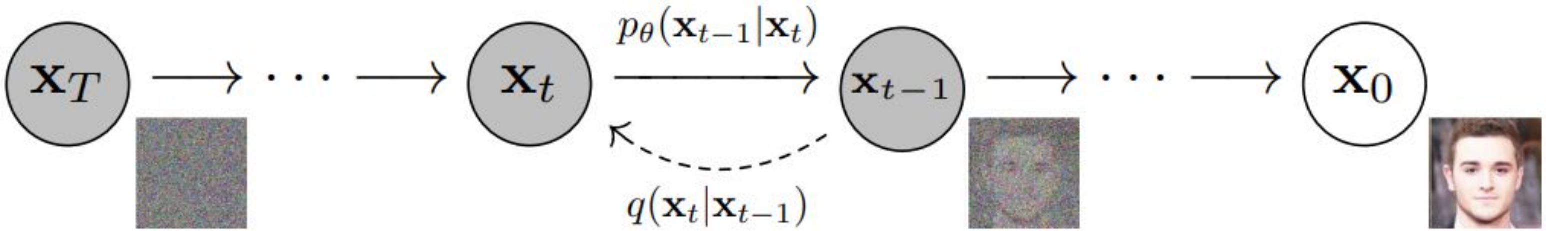
Jenish Bajracharya (jbajracharya@umass.edu),
Shivam Raj (shivamraj@umass.edu),
Suvid Sahay (suvidsahay@umass.edu)

Project #54

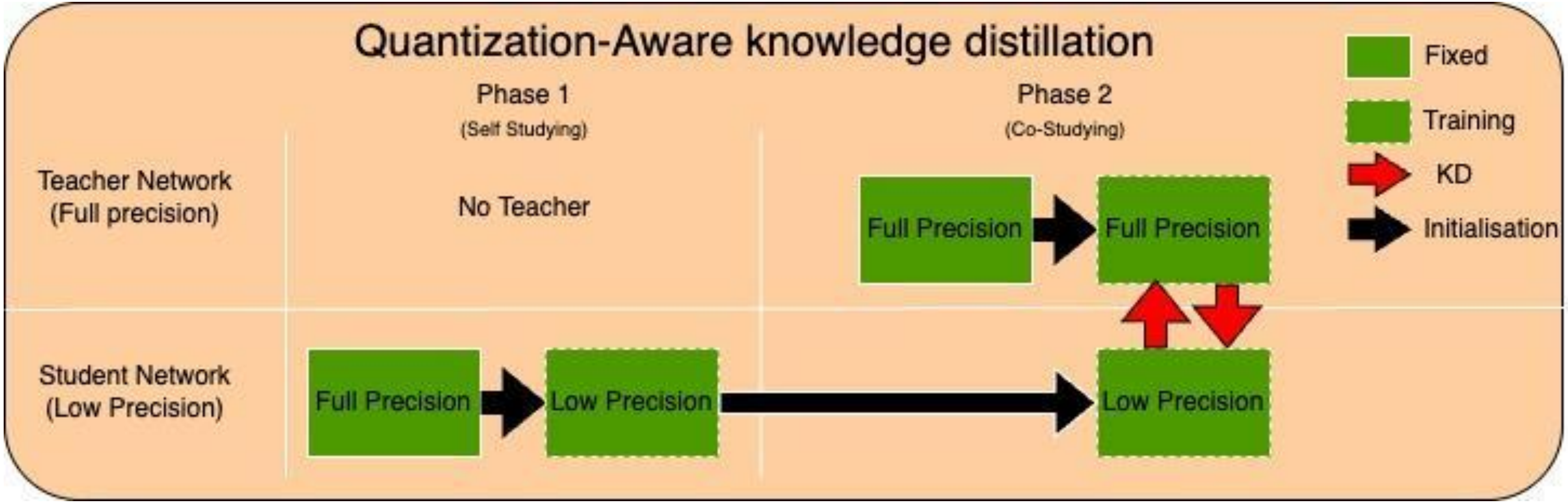
Diffusion Models

Used to generate image from noise

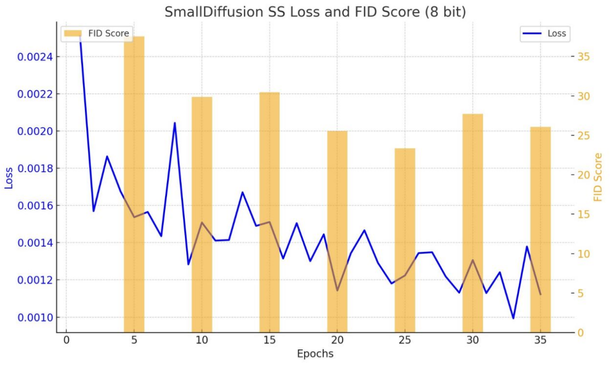


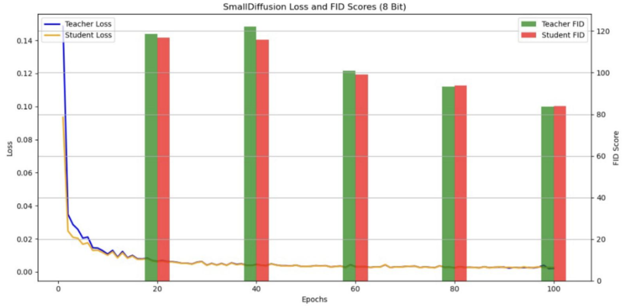

$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$\mathbf{x}_{t-1}$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$
$\mathbf{x}_0$

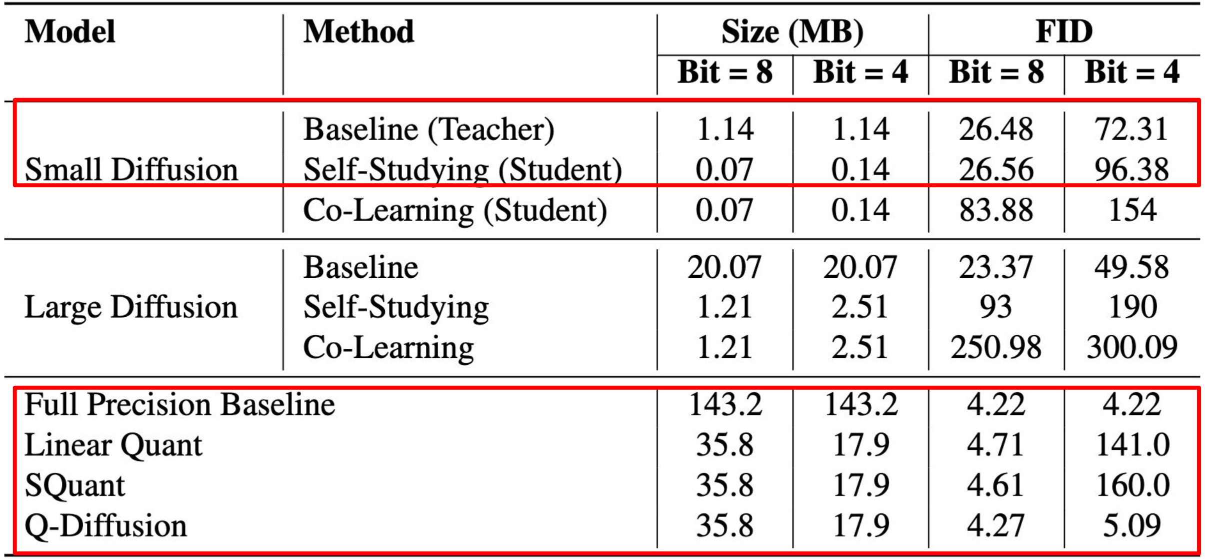
Quantization-Aware Knowledge Distillation



Results and Conclusion







Model	Method	Size (MB)		FID	
		Bit = 8	Bit = 4	Bit = 8	Bit = 4
Small Diffusion	Baseline (Teacher)	1.14	1.14	26.48	72.31
	Self-Studying (Student)	0.07	0.14	26.56	96.38
	Co-Learning (Student)	0.07	0.14	83.88	154
Large Diffusion	Baseline	20.07	20.07	23.37	49.58
	Self-Studying	1.21	2.51	93	190
	Co-Learning	1.21	2.51	250.98	300.09
Full Precision Baseline		143.2	143.2	4.22	4.22
Linear Quant		35.8	17.9	4.71	141.0
SQuant		35.8	17.9	4.61	160.0
Q-Diffusion		35.8	17.9	4.27	5.09

Compact Diffusion Model

Jenish Bajracharya, Suvid Sahay and Shivam Raj

Exploring Noise Schedulers in Diffusion Models

Pranav Balakrishnan (pranavbalakr@umass.edu),
Sidisha Barik (sbarik@umass.edu)

Project #56

- **Objective:**
  Investigate the impact of **noise schedulers** on training and sampling in diffusion models across different formulations (i.e. VP, Sub-VP).
- **Focus:**
  Evaluate how noise schedulers influence the noise levels the model prioritizes during training and the resulting image quality.
- **Why Noise Scheduling Matters:**
  - Noise schedulers dictate how noise is introduced during training, directly affecting model focus and performance.
  - Noise schedulers are critical for controlling **training dynamics** and **image quality** in diffusion models.
  - Balancing noise levels during training can lead to **higher-quality image generation**, requiring an optimal trade-off between structural fidelity and texture detail.
- **Human Analogy:**
  Similar to human perception, diffusion models can benefit from prioritizing specific frequencies for better results:
  - **Low frequencies:** Handle higher noise levels, aiding general structure.
  - **High frequencies:** Improve texture and detail, reducing blur.

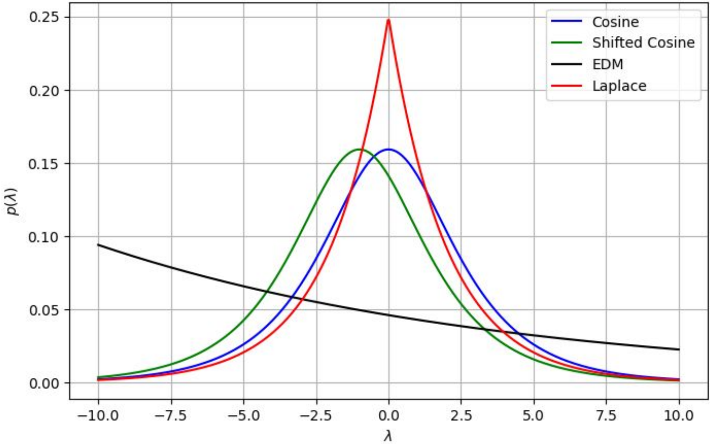
Photo credit: kipply

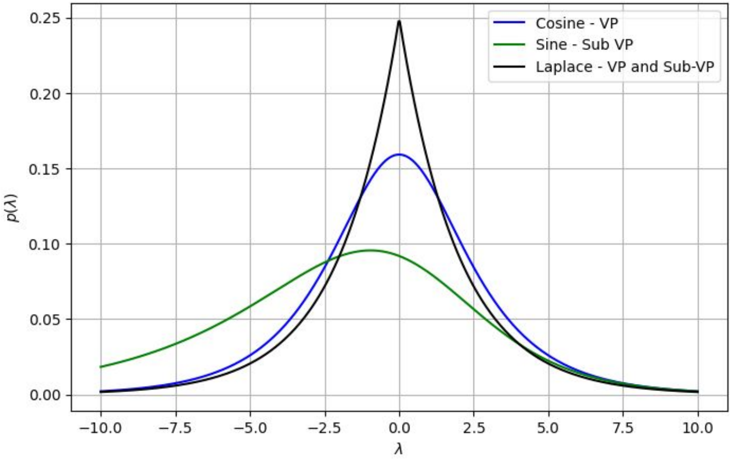
	$p(\lambda)$
Cosine VP	$\operatorname{sech}\left(\frac{s\lambda}{2}\right) \cdot \frac{s}{2\pi}$
Laplace VP and Sub-VP	$\frac{1}{2b} \exp\left(-\frac{\lvert\lambda-\mu\rvert}{b}\right)$
Sine Sub-VP	$\frac{1}{\pi} \cdot \left(\frac{e^{\lambda/2}}{\sqrt{e^{\lambda}+2e^{\lambda/2}}}\right) \cdot \frac{1}{1+e^{\lambda/2}}$

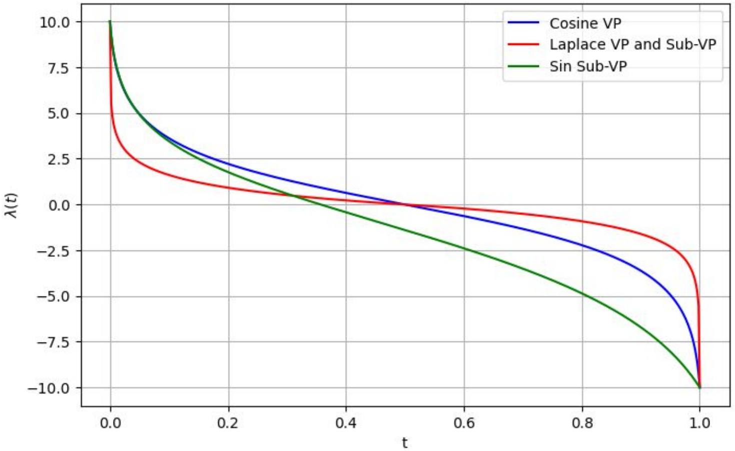
Table 1. $p(\lambda)$ for different Noise Schedulers under different formulations

	$\lambda(t)$
Cosine VP	$\frac{-2}{s} \log \tan\left(\frac{\pi t}{2}\right)$
Laplace VP and Sub-VP	$\mu - b \operatorname{sgn}(0.5 - t) \log\left(1 - 2\lvert t - 0.5\rvert\right)$
Sine Sub-VP	$2 \log\left((1/\sin\frac{\pi t}{2}) - 1\right)$

Table 2. $\lambda(t)$ for different Noise Schedulers under different formulations







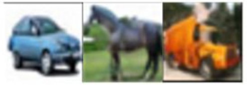


Figure 8. Cosine (s = 0.75) VP

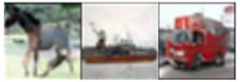


Figure 9. Cosine (s = 1) VP

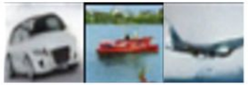


Figure 10. Cosine (s = 2) VP

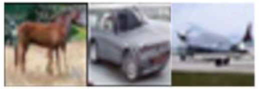


Figure 14. Sine (s = 1) Sub VP

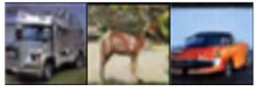


Figure 15. Sine (s = 2) Sub VP

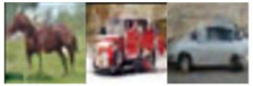


Figure 16. Sine (s = 3) Sub VP

Key Findings

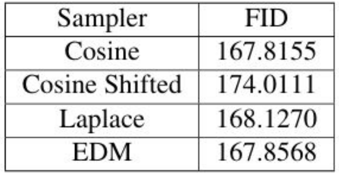
- **Best Samplers:** Cosine and EDM performed best on the baseline Cosine VP model in terms of FID-3k and visual quality.
- **Best Models:**
  - Baseline Cosine model (VP formulation)
  - Sine (s=1) model (Sub VP formulation)

Observations

- **Laplace Scheduler:** Recent work (Hang et al., 2024) shows it outperforms the Cosine scheduler by focusing more aggressively on specific noise levels.
- **Our Findings:** Aggressive focus on mid-range noise levels is generally not helpful for VP and Sub-VP formulations.

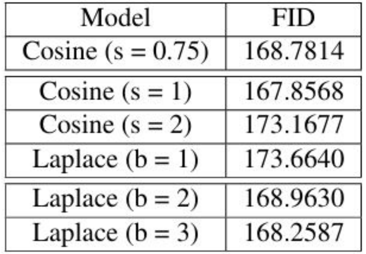
Limitations

- **Compute Constraints:**
  - FID scores calculated with only **3k samples** (vs. 50k in similar studies).
  - Models trained for **25k iterations** (vs. millions).
- Further training and evaluation are needed for concrete conclusions.



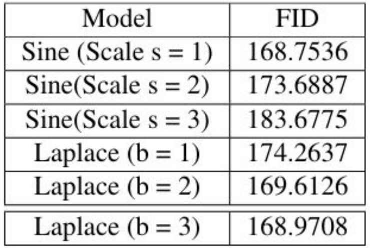
Sampler	FID
Cosine	167.8155
Cosine Shifted	174.0111
Laplace	168.1270
EDM	167.8568

Table 5. FID-3k scores observed for different samplers using samples generated using baseline Cosine Noise Scheduler model in VP formulation



Model	FID
Cosine (s = 0.75)	168.7814
Cosine (s = 1)	167.8568
Cosine (s = 2)	173.1677
Laplace (b = 1)	173.6640
Laplace (b = 2)	168.9630
Laplace (b = 3)	168.2587

Table 6. FID-3K Score Comparison for models trained in VP Formulation



Model	FID
Sine (Scale s = 1)	168.7536
Sine(Scale s = 2)	173.6887
Sine(Scale s = 3)	183.6775
Laplace (b = 1)	174.2637
Laplace (b = 2)	169.6126
Laplace (b = 3)	168.9708

Table 7. FID-3K Score Comparison for models trained in Sub-VP Formulation

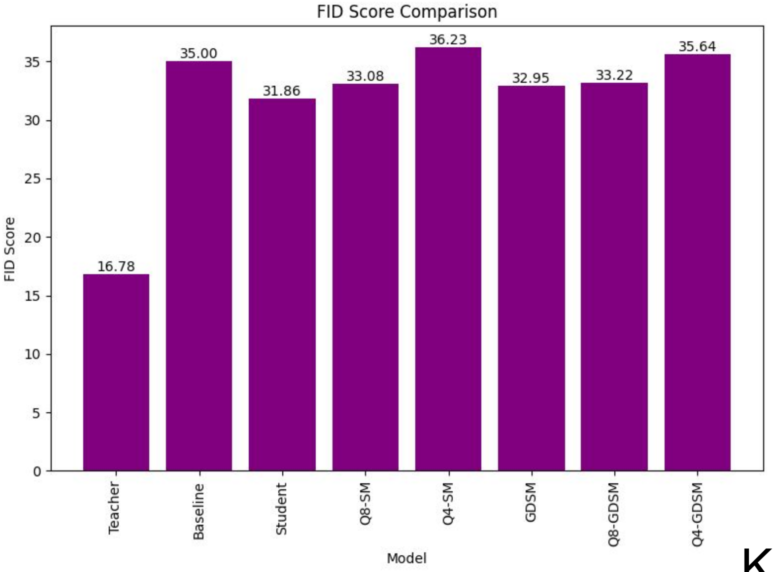
Compact diffusion models for Cifar-10

Shreya Birthare (sbirthare@umass.edu),
Tirth Bhagat (tbhagat@umass.edu)

Project #57

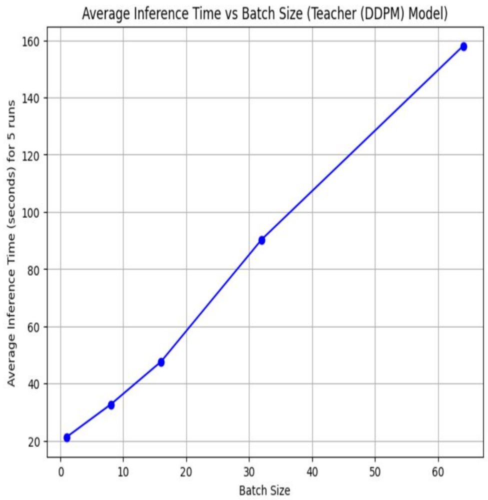
MOTIVATION AND AIM:

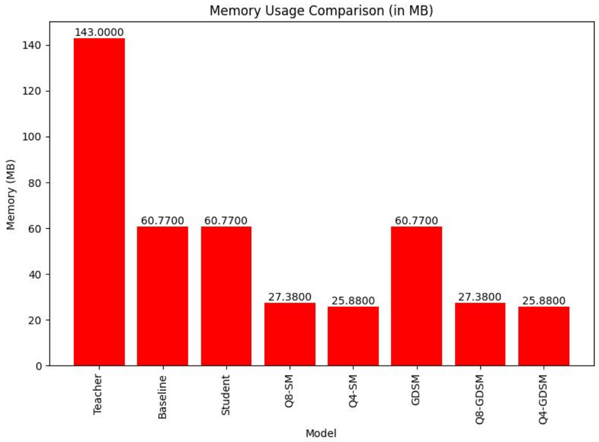
- Diffusion models take up alot of computational resources in terms of both memory and time and is expensive.
- Aim to make the models less computationally heavy and expensive to deploy it on resource constrained environments.
- Use Knowledge Distillation and Quantization
- Generate new images using the distilled and quantized models and evaluate them
- Evaluate using FID Score, Model Size, Number of Parameters, FLOPs, Inference Time



Teacher

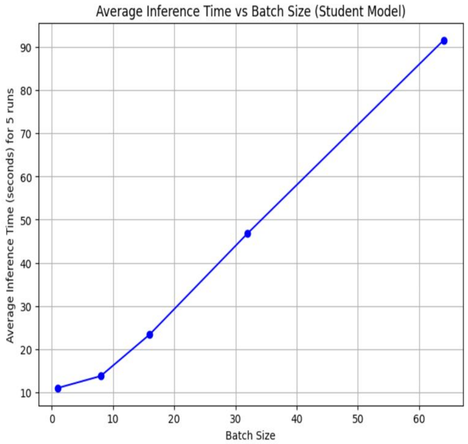
Baseline


Average Inference Time vs Batch Size (Teacher (DDPM) Model)
Average Inference Time (seconds) for 5 runs
Batch Size
20
40
60
80
100
120
140
160
0
10
20
30
40
50
60



Knowledge
Distilled Student

8Q-SM


Average Inference Time vs Batch Size (Student Model)
Average Inference Time (seconds) for 5 runs
Batch Size
10
20
30
40
50
60
70
80
90
0
10
20
30
40
50
60

CONCLUSION

- Larger models allow for better image generation but at the expense of time and memory.
- Decreasing the model architecture or quantization can reduce the model memory consumption and generate images faster but at a cost of lower quality.
- Knowledge distillation can help smaller models create better images by having a teacher model help guide them to learn the ”good features” while maintaining the benefits of having a smaller model.
- Quantization can help decrease model size and speed up inference time at the cost of poorer images

Compact diffusion models for Cifar-10

Atif Abedeen (aabedeen@umass.edu),
Darsh Gondalia (dgondalia@umass.edu)

Project #59

Project Overview:

Lite-Diffusion (L-Diff) addresses the computational demands of diffusion models, enabling rapid and efficient image generation on resource-constrained devices through various model compression techniques.

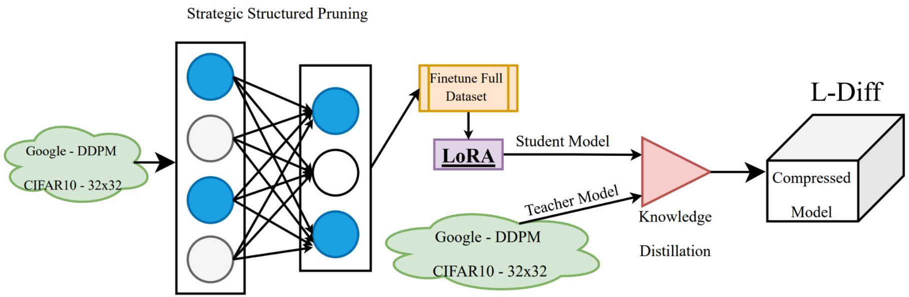
Motivation:

Edge devices, such as smartphones and IoT platforms, face significant challenges when running full-sized Denoising Diffusion Probabilistic Models (DDPMs) due to limited memory and computational resources.

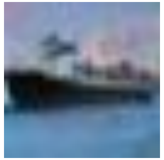
Background on Techniques Explored for Model Compactness:

- Pruning: Reduces redundant model parameters to save memory and computation.
- Quantization: Lowers precision of weights for smaller, faster models.
- Knowledge Distillation: Transfers knowledge from large models to smaller ones.
- Low-Rank Adaptation (LoRA): Efficiently fine-tunes models using low-rank updates.

This project focuses on strategically combining these compression techniques to create a compact diffusion model that retains most of its core performance and value.



Some images generated by L-Diff



Key Findings and Results:

Structured Pruning:

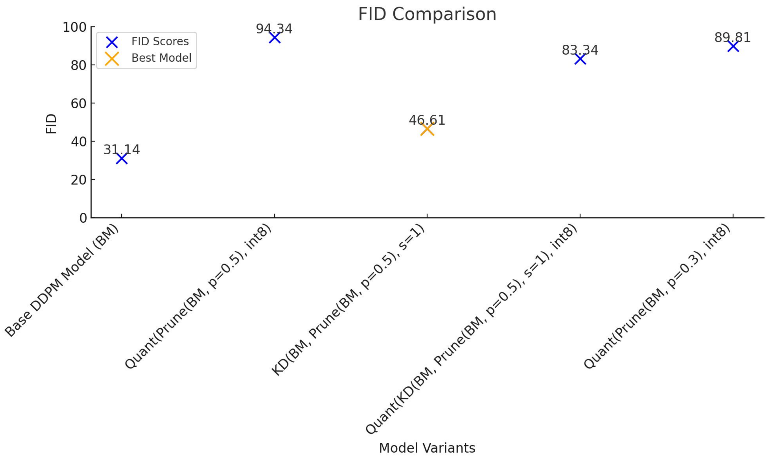
- Pruned 50% of the neurons in the convolutional and attention blocks
- Preserved Time Embedding layers to capture necessary temporal information
- Reduced model size by up to **60%** with minimal performance loss.

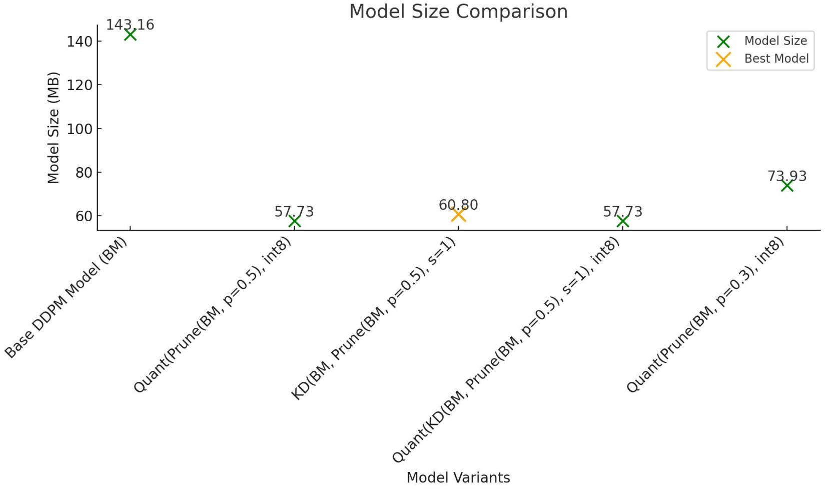
Best Configuration: 50% pruned DDPM model + Fine-Tuning using KD

- **Model Size:** Reduced from 144 MB to 60.8 MB
- **FID:** 31.14 to 46.61
- **GMACs:** Reduced from 6.06 to 2.13

Comparison to Previous Work:

- Previous work aimed to use these model compression techniques in isolation
- Implemented Structured Pruning along with dynamic quantization, Low Rank adaptation, and further Fine-tuned with Knowledge Distillation





Conclusion

Effective Compression of DDPMs for Edge Devices:

- Explored various model compression techniques to optimize DDPMs for CIFAR-10.
- Aimed to deploy models on resource-constrained edge devices.

Key Takeaway:

- Strategic structured pruning, complemented by fine-tuning and knowledge distillation, is the most effective approach for compressing diffusion models.
- Strikes an optimal balance between reducing model complexity and retaining high-quality image generation.
- Makes advanced generative models more accessible for practical, real-world applications on edge devices.

Future Research Directions:

- Focus on other strategic quantization techniques like Quantization Aware Fine-Tuning
- Further continue work on refining Low Rank Adaptation technique and also apply on attention blocks
- Work with datasets with larger image sizes like imageNet

Text augmentation in LLMs

Ajith Krishna Kanduri (akanduri@umass.edu),
Spoorthi Siri Malladi (smalladi@umass.edu)

Project #64

Motivation,Background and Summary

Motivation:

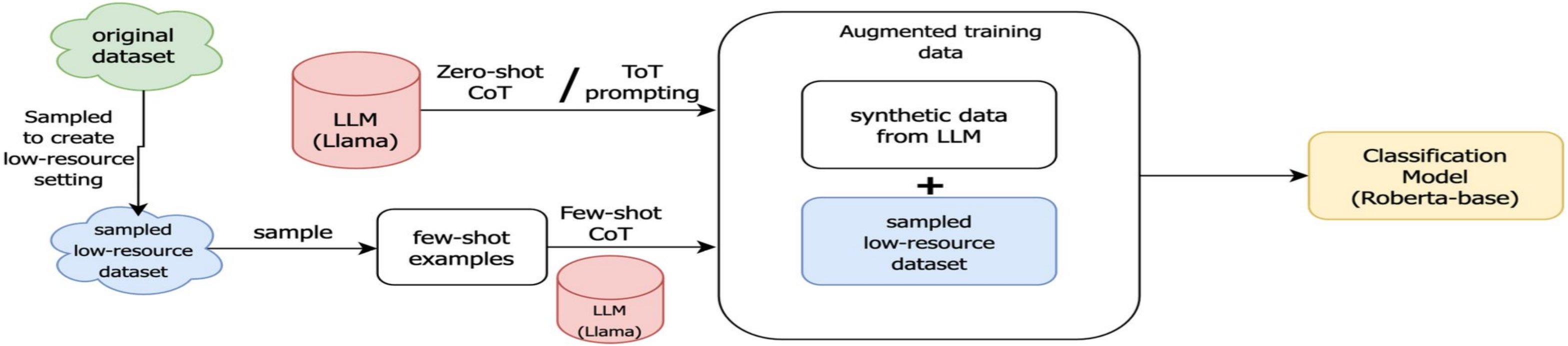
- Improve synthetic data generation for NLP tasks like sentiment analysis.
- Address gaps in effectiveness between simple and complex

Background:

- Text classification is a key challenge in NLP with wide applications.
- Prompt engineering leverages LLMs to create high-quality

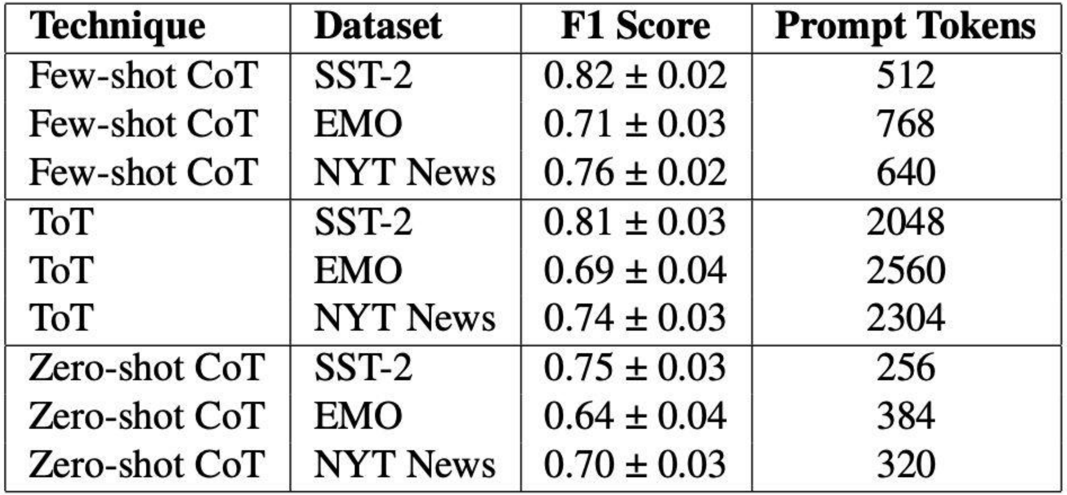
Summary:

- Focuses on prompt engineering strategies for text classification using LLMs.
- Evaluates methods like Zero-shot CoT, Few-shot

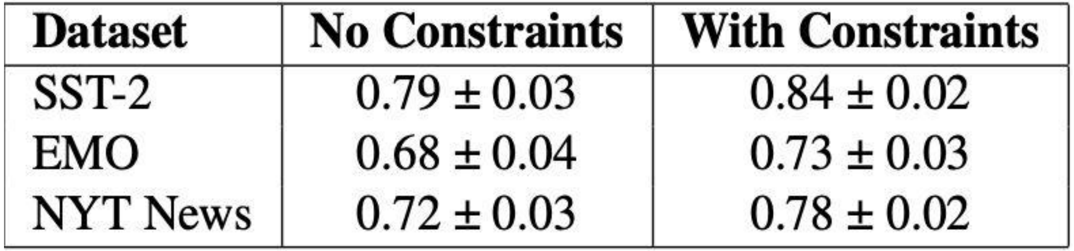

original dataset
Sampled to create low-resource setting
sampled low-resource dataset
sample
few-shot examples
Few-shot CoT
LLM (Llama)
LLM (Llama)
Zero-shot CoT / ToT prompting
Augmented training data
synthetic data from LLM
+
sampled low-resource dataset
Classification Model (Roberta-base)

Results and Comparison

- Few-shot CoT improves accuracy and data quality significantly.
- Zero-shot CoT performs reasonably but lacks robustness.
- Tree-of-Thoughts shows minimal improvement despite added complexity
- The incorporation of domain-specific constraints and contextual examples on Few-shot CoT proved crucial in enhancing the quality of synthetic data



Technique	Dataset	F1 Score	Prompt Tokens
Few-shot CoT	SST-2	0.82 ± 0.02	512
Few-shot CoT	EMO	0.71 ± 0.03	768
Few-shot CoT	NYT News	0.76 ± 0.02	640
ToT	SST-2	0.81 ± 0.03	2048
ToT	EMO	0.69 ± 0.04	2560
ToT	NYT News	0.74 ± 0.03	2304
Zero-shot CoT	SST-2	0.75 ± 0.03	256
Zero-shot CoT	EMO	0.64 ± 0.04	384
Zero-shot CoT	NYT News	0.70 ± 0.03	320



Dataset	No Constraints	With Constraints
SST-2	0.79 ± 0.03	0.84 ± 0.02
EMO	0.68 ± 0.04	0.73 ± 0.03
NYT News	0.72 ± 0.03	0.78 ± 0.02

Conclusion and Future Work

Conclusion:

- Few-shot CoT is the most effective for generating synthetic data.
- Simpler, structured prompts outperform overly complex techniques.
- Prompt design should prioritize clarity and focus over complexity.

Future Work:

- Integrate domain-specific knowledge to improve relevance.
- Test these strategies on additional NLP tasks and benchmarks.
- Investigate hybrid approaches combining simple and complex methods.

UMassAmherst | Manning College of Information & Computer Sciences

SyntheticHate

Data Augmentation Methods for Hate Speech Classification

Ayush Gupta (ayushanilgup@umass.edu),
Debrup Das (debrupdas@umass.edu),
Soumitra Das (soumitradas@umass.edu)

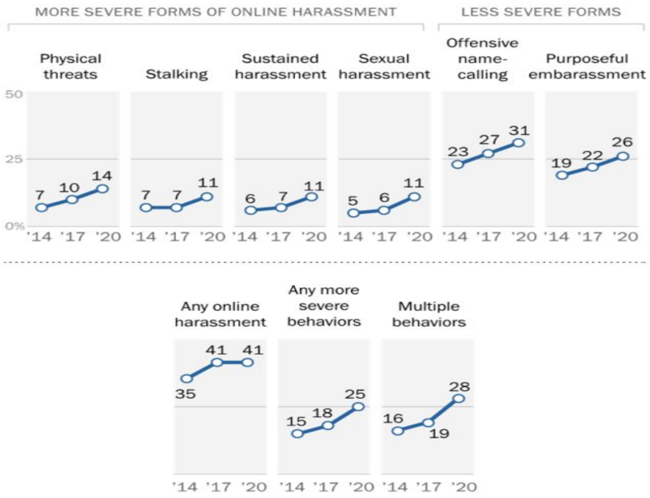
Project #67

Introduction

Objective: Improve the performance of hate speech classification by augmenting an imbalanced dataset using both traditional and LLM-based methods to generate synthetic hate speech examples.

Motivation:

- Hate speech is a pressing issue online, requiring robust automated detection systems.
- Training effective machine learning models is challenging due to limited and imbalanced datasets.
- Effective detection systems can help reduce the spread of harmful content, fostering safer and more inclusive online communities.



Compared with 2017, similar share of Americans have experienced any type of online harassment – but more severe encounters have become more common

% of U.S. adults who say they have personally experienced the following behaviors online

Note: Those who did not give an answer are not shown.
Source: Survey of U.S. adults conducted Sept. 8-13, 2020.
"The State of Online Harassment"

PEW RESEARCH CENTER

Methods and Results

- **Typo Injection**
  - Fine-tuned BART model using Github Typo Corpus.
  - Used gpt-3.5 & gpt-4o to generate augmented data with varying number of typos
- **Backtranslation**
  - Used translation models of 5 languages from MarianMT framework.
  - Performed sequential back-and-forth translations by prompting gpt-4o.
  - Employed gpt-4o for executing backtranslation with a direct prompt of batch data.
- **Explanation Augmentation**
  - Used few-shot prompting with in-context examples using Llama-3.1-8B to generate explanations for each example.
  - Added newly generated rationales to the existing dataset to improve LLM's reasoning associated with hate detection.
  - Performed fine-tuning with LORA on LLM based classifier.

Methods	**Accuracy**	**F1-score (hate class)**
Original data	0.76	0.69
Typo-injection (traditional)	0.82	0.79
Backtranslation (traditional)	0.80	0.79
Typo-injection (LLM)	0.79	0.75
Backtranslation (LLM)	**0.83**	**0.80**
Explanation augmentation	0.61	0.55

Conclusion

- Classical methods like typo injection and backtranslation, using fine-tuned transformer models, effectively improve data diversity and address class imbalance.

- We observe that performing back-translation using GPT-4o through instruction prompting results in substantial improvements. Similar gains are also observed using gpt-3.5-turbo for typo generation.

- The performance of explanation augmentation sufferers heavily from the highly imbalanced nature of the dataset.

- LLM-based approaches require significant computational resources, making them less accessible for practitioners with limited budgets.

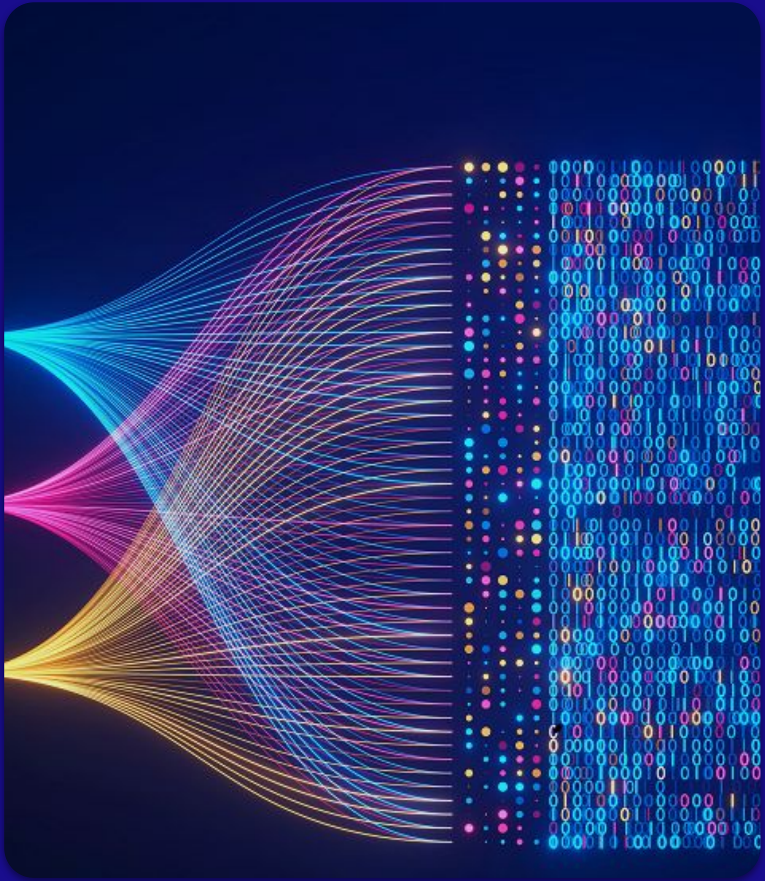
Project #68

CS682

LLMs vs Established Text Augmentation Techniques for Classification (Group 68)

Thomas Ji (tji@umass.edu)
Andrew Lin (andrewlin@umass.edu)
Kevin Oliveira Downing (kjdowning@umass.edu)

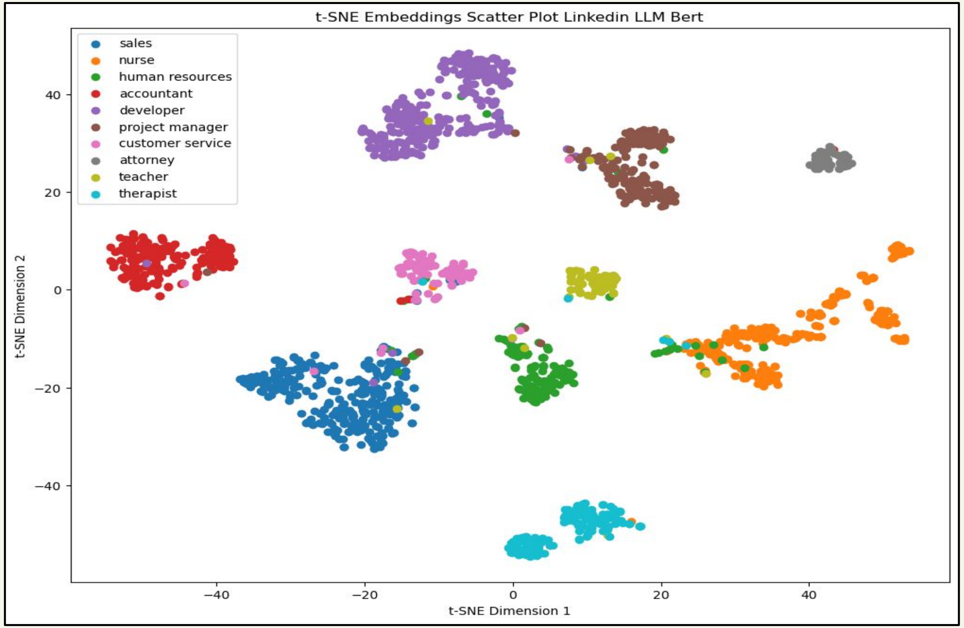
12/03/2025



LinkedIn Results

The LinkedIn dataset was classifying job titles from description.

- Small class imbalance, with many job-related keywords
- Minimal difference between augmentation methods (and no augmentation)
- Clear separation implies that the model is already close to peak performance without augmentation

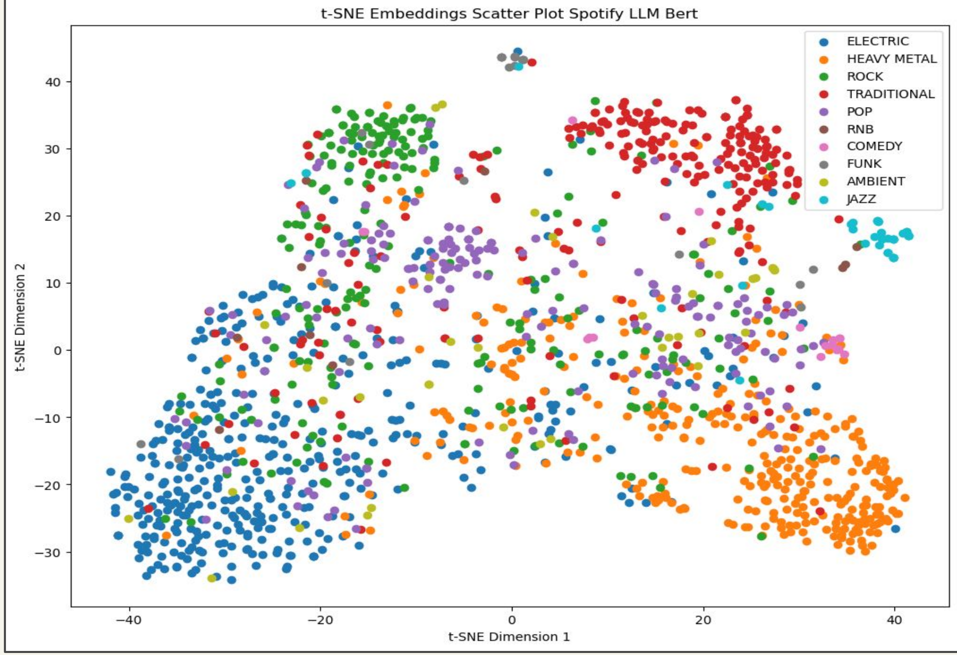


	Linkedin		
	Accuracy	Macro F1	Weighted F1
NA	0.96	**0.96**	**0.96**
Classical	**0.97**	0.96	0.96
BT	0.96	0.95	0.96
LLM	0.96	**0.96**	**0.96**

Spotify Results

The Spotify dataset was about classifying song lyrics with genres.

- Large class imbalance
- LLM and Backtranslation improved F1 scores significantly
- Demonstrates that textual augmentation methods are very relevant



	Spotify		
	Accuracy	Macro F1	Weighted F1
NA	0.64	0.45	0.63
Classical	**0.65**	0.54	0.63
BT	0.64	**0.58**	0.64
LLM	**0.65**	0.57	**0.65**

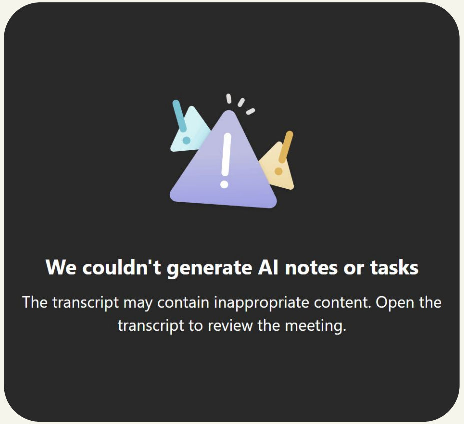
LLM Summary

“Rewrite the following [type of data (job postings or song lyrics)] by making small changes: [data].”

Although LLM performed the best out of the text augmentation methods, the difference is minimal and there are some important drawbacks to consider.

Expensive & Time costly

Most LLMs are reliant on APIs which can prove quite expensive and time consuming over large batches of data.



We couldn't generate AI notes or tasks

The transcript may contain inappropriate content. Open the transcript to review the meeting.

Inappropriate content filter

Augmentation of datasets that contain inappropriate content can prove problematic as many APIs like Gemini refuse to answer prompts with such language.

Textual augmentation for LLMs

Aminta Rebecca Asheel (aasheel@umass.edu),
Muskan Kothari (mkothari@umass.edu)

Project #69

Introduction

Motivation & background

Challenge: Class imbalance in emotion classification leads to biased predictions.

Approach: Improved classifier robustness through synthetic data generation.

Techniques used:

- Traditional: TextAttack
- LLM-Based: GPT-Neo

Objective

Objective: Evaluated LLM-based augmentation vs. traditional techniques for imbalanced emotion classification.

Focus: Analyzed the impact of contextual prompts on data augmentation effectiveness.

Comparison to previous work

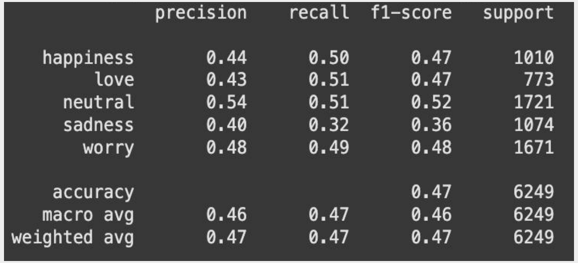
- **Traditional TextAttack:** Improved class representation but lacked nuance and context-rich samples.
- **LLM-Based Augmentation:** Generated diverse, semantically aligned samples, addressing rule-based limitations and outperforming traditional methods.

Results

Prompt Engineering Process:

- **Initial Approach:** Used raw training data samples as seeds for generation but yielded limited diversity.
- **Refined Approach:** Augmented data by providing both the emotion label and a representative sample to guide generation.
- **Final Approach:**
  - Contextualized prompts by defining the emotion underlying each sentiment (e.g., sadness reflects loss and regret).
  - Generated three diverse samples per prompt using GPT-Neo.
  - Validated outputs with a secondary classifier (cardiffnlp/twitter-roberta-base-sentiment) to ensure alignment with the target sentiment.

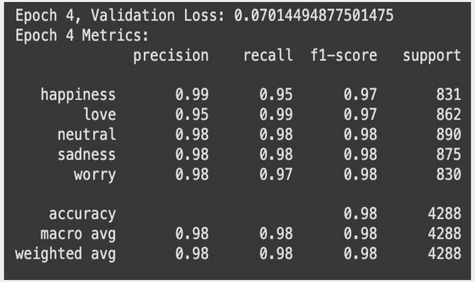
- **Baseline Model:** Accuracy: 47%; significant bias against minority classes (*love*, *sadness*).
- **TextAttack Augmentation:** Accuracy improved to ~66%, with gains in recall for minority classes but limited sample diversity.
- **LLM-Based Augmentation:**
  - Accuracy improved to 71% post GPT-Neo augmentation without context
  - With context, the model performed with an accuracy as good as 98%
  - Recall for *love* increased from 39% (baseline) to 99%; *sadness* rose from 54% to 98%.



```
              precision    recall  f1-score   support

   happiness       0.44      0.50      0.47      1010
        love       0.43      0.51      0.47       773
     neutral       0.54      0.51      0.52      1721
     sadness       0.40      0.32      0.36      1074
       worry       0.48      0.49      0.48      1671

    accuracy                           0.47      6249
   macro avg       0.46      0.47      0.46      6249
weighted avg       0.47      0.47      0.47      6249
```

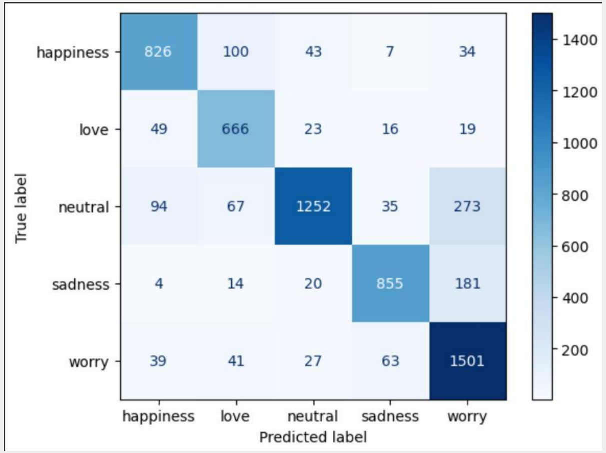


```
Epoch 4, Validation Loss: 0.07014494877501475
Epoch 4 Metrics:
              precision    recall  f1-score   support

   happiness       0.99      0.95      0.97       831
        love       0.95      0.99      0.97       862
     neutral       0.98      0.98      0.98       890
     sadness       0.98      0.98      0.98       875
       worry       0.98      0.97      0.98       830

    accuracy                           0.98      4288
   macro avg       0.98      0.98      0.98      4288
weighted avg       0.98      0.98      0.98      4288
```

Conclusion



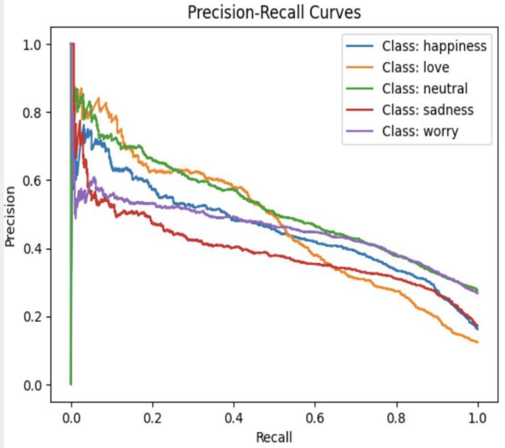
True label \ Predicted label	happiness	love	neutral	sadness	worry
happiness	826	100	43	7	34
love	49	666	23	16	19
neutral	94	67	1252	35	273
sadness	4	14	20	855	181
worry	39	41	27	63	1501

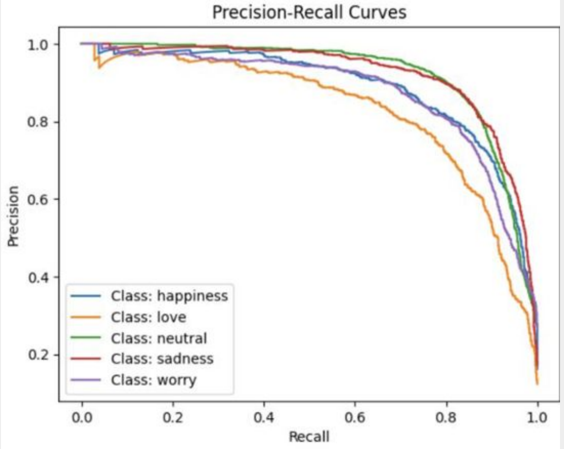
Key Findings:

- LLM-based augmentation effectively mitigated class imbalance, improved accuracy, and enhanced minority class metrics like recall and F1-scores.
- LLM-generated samples provided high-quality, contextually rich data, enabling better generalization and reducing bias in predictions.

Impact:

- Highlights the pivotal role of advanced data augmentation techniques in addressing class imbalance, positioning LLMs as essential tools for modern NLP tasks.





LLMs to the Rescue: Can Text Style Transfer bridge the Gap in Minority Class Detection?

Aashnna Soni (aashnnasoni@umass.edu),
Fabeha Fabeha Fatima (ffatima@umass.edu)

Project #72

LLMs to the Rescue

MOTIVATION

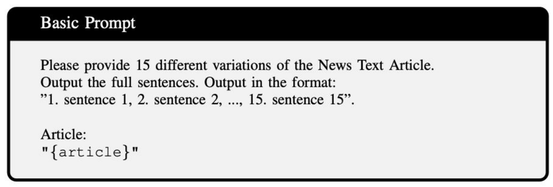
Class imbalance is a critical challenge in text classification, leading to biased predictions and suboptimal performance, particularly for minority classes. This issue is especially concerning in high-stakes fields like healthcare, finance, and news categorization, where errors can have serious consequences.

SUMMARY

Baseline Model: Generates simple variations of news articles.
Advanced Model: Applies **style transfer** to create text variations across distinct journalistic styles (e.g., Investigative, Editorial, Breaking News).

Classifiers Used:
RoBERTa: Transformer-based model.
Logistic Regression: More interpretable and computationally efficient.

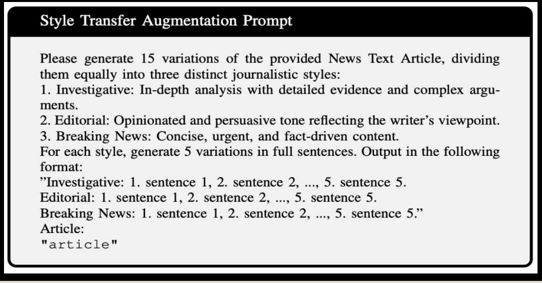
Key Findings:
Style transfer and LLM-based augmentation significantly **improve classification accuracy**, especially for **minority classes.**
The generated text samples are **high-quality** and **diverse**, addressing class imbalance effectively.



Basic Prompt

Please provide 15 different variations of the News Text Article.
Output the full sentences. Output in the format:
"1. sentence 1, 2. sentence 2, ..., 15. sentence 15".

Article:
"{article}"

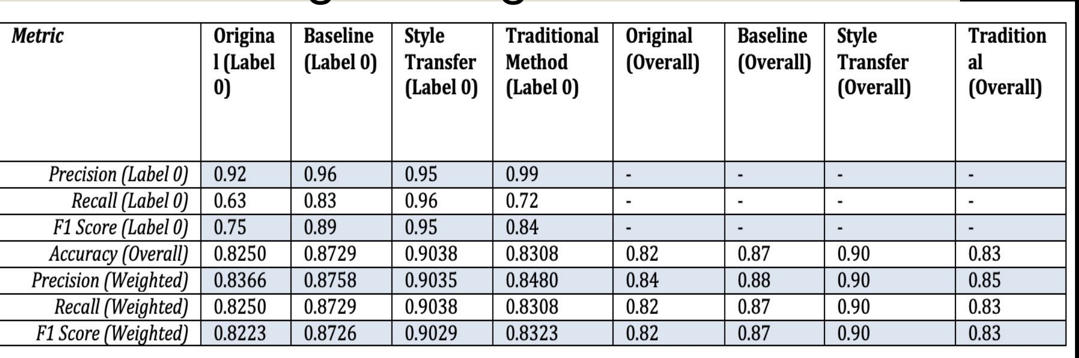


Style Transfer Augmentation Prompt

Please generate 15 variations of the provided News Text Article, dividing them equally into three distinct journalistic styles:
1. Investigative: In-depth analysis with detailed evidence and complex arguments.
2. Editorial: Opinionated and persuasive tone reflecting the writer's viewpoint.
3. Breaking News: Concise, urgent, and fact-driven content.
For each style, generate 5 variations in full sentences. Output in the following format:
"Investigative: 1. sentence 1, 2. sentence 2, ..., 5. sentence 5.
Editorial: 1. sentence 1, 2. sentence 2, ..., 5. sentence 5.
Breaking News: 1. sentence 1, 2. sentence 2, ..., 5. sentence 5."
Article:
"article"

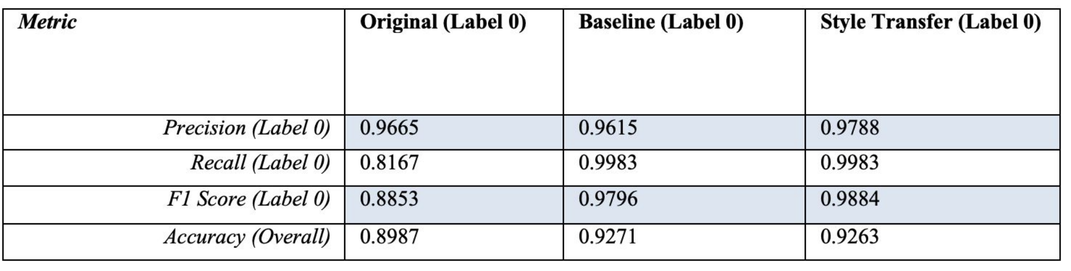
Impact:
The results show that LLM-based augmentation offers a **scalable solution** for tackling class imbalance in NLP tasks, improving both fairness and model performance

Logistic Regression Results

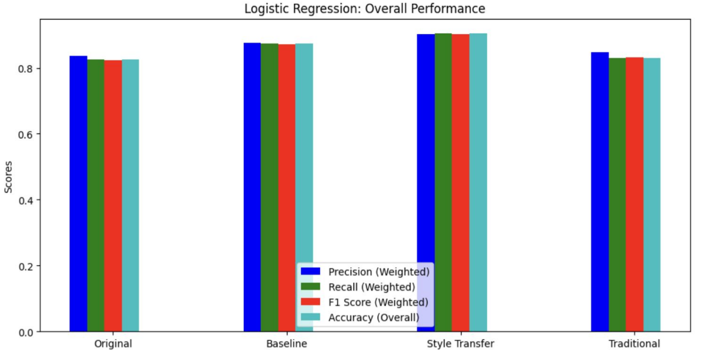


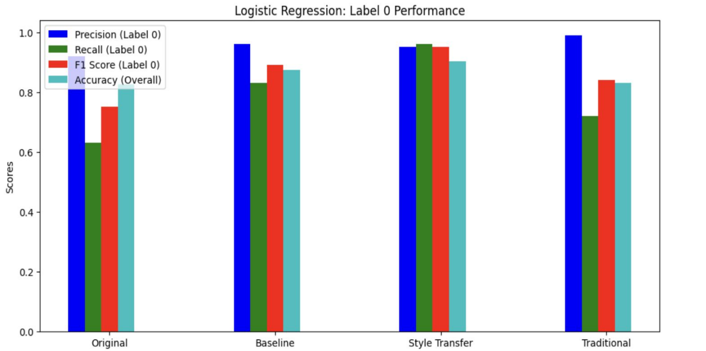
Metric	Original (Label 0)	Baseline (Label 0)	Style Transfer (Label 0)	Traditional Method (Label 0)	Original (Overall)	Baseline (Overall)	Style Transfer (Overall)	Traditional (Overall)
Precision (Label 0)	0.92	0.96	0.95	0.99	-	-	-	-
Recall (Label 0)	0.63	0.83	0.96	0.72	-	-	-	-
F1 Score (Label 0)	0.75	0.89	0.95	0.84	-	-	-	-
Accuracy (Overall)	0.8250	0.8729	0.9038	0.8308	0.82	0.87	0.90	0.83
Precision (Weighted)	0.8366	0.8758	0.9035	0.8480	0.84	0.88	0.90	0.85
Recall (Weighted)	0.8250	0.8729	0.9038	0.8308	0.82	0.87	0.90	0.83
F1 Score (Weighted)	0.8223	0.8726	0.9029	0.8323	0.82	0.87	0.90	0.83

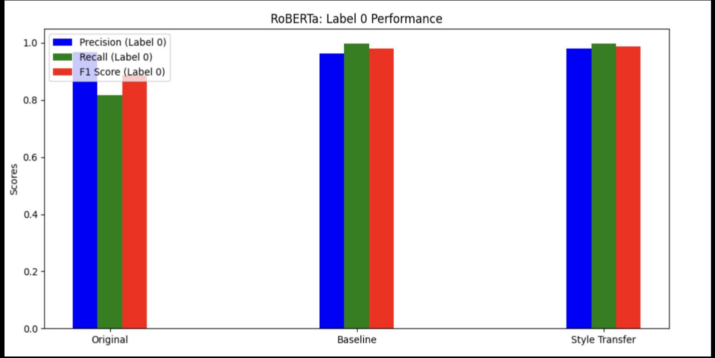
RoBERTa Results



Metric	Original (Label 0)	Baseline (Label 0)	Style Transfer (Label 0)
Precision (Label 0)	0.9665	0.9615	0.9788
Recall (Label 0)	0.8167	0.9983	0.9983
F1 Score (Label 0)	0.8853	0.9796	0.9884
Accuracy (Overall)	0.8987	0.9271	0.9263







Conclusion:

- **Better Minority Class Detection**: Our approach improves classification accuracy, especially for underrepresented classes.

- **Addresses Class Imbalance**: LLM-based augmentation generates diverse samples, enhancing fairness and robustness in NLP tasks.

- **Scalable Solution**: The method is adaptable to various NLP applications, from news categorization to healthcare and finance.

- **Potential for Future Use**: This work opens opportunities for applying LLM augmentation to other NLP challenges, such as sentiment analysis and fraud detection.

Exploring GAN-based Augmentation Technique for Improving Classification Accuracy of Medical Images

Anisha Prajapati (anishaprajap@umass.edu),
Geetanjali Aich (gaich@umass.edu)

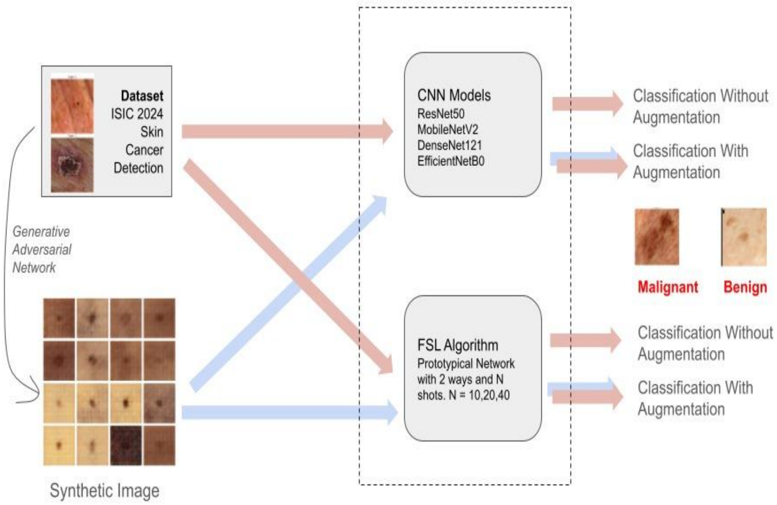
Project #74

Problem Statement

Investigate whether GAN-based data augmentation can improve accuracy, IoU, and other metrics for CNN and few-shot learning models on medical image classification. Specifically, produce augmented images with DCGAN and evaluate classification performance using CNNs and Prototypical Networks.
Dataset Used: ISIC 2024 Skin Cancer Detection Dataset

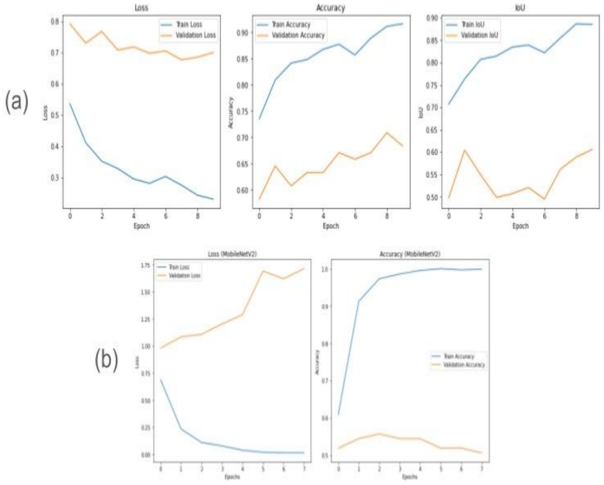
Approach

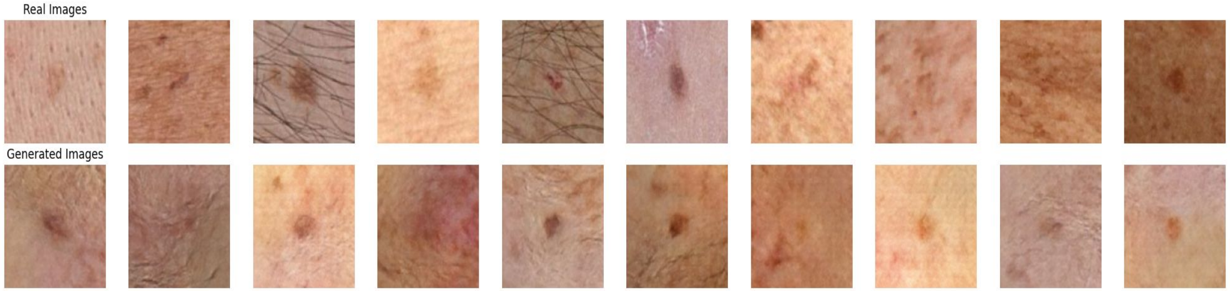
1. Implementation of DCGAN architecture with generator and discriminator networks
2. Generator: 100-dimensional latent vector to 128×128×3 images through transposed convolutions
3. Discriminator: Processing images through convolutional layers with dropout and LeakyReLU
4. Monitoring system to track diversity and prevent mode collapse
5. Training with Adam optimizer (learning rate=0.0002, β=0.5) for 40 epochs
6. Comprehensive metrics tracking including generator/discriminator losses
7. Visual inspection and comparison with real medical images
8. Integration with classical CNN models and Prototypical Network for FSL


Dataset
ISIC 2024 Skin Cancer Detection
Generative Adversarial Network
Synthetic Image
CNN Models
ResNet50
MobileNetV2
DenseNet121
EfficientNetB0
FSL Algorithm
Prototypical Network with 2 ways and N shots. N = 10,20,40
Classification Without Augmentation
Classification With Augmentation
Malignant
Benign
Classification Without Augmentation
Classification With Augmentation

Results

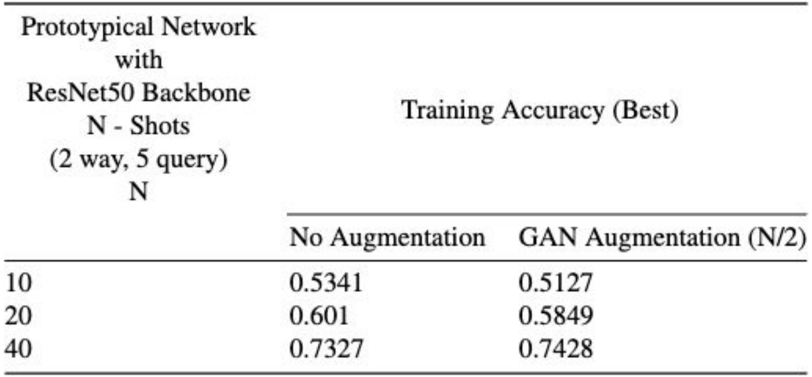
1. DenseNet121 with DCGAN augmentation achieved best test accuracy of 83.55%
2. Significant improvement in IoU scores across models with augmentation
3. DenseNet121 IoU improved from 0.51 to 0.6833
4. MobileNetV2 IoU improved from 0.3282 to 0.5676
5. Prototypical Network showed marginal improvement for 40-shot 2-way 5-query setup
6. Stable GAN training with balanced generator/discriminator performance
7. Generated images maintained key characteristics of skin lesions
8. Consistent improvement in image quality compared to previous models





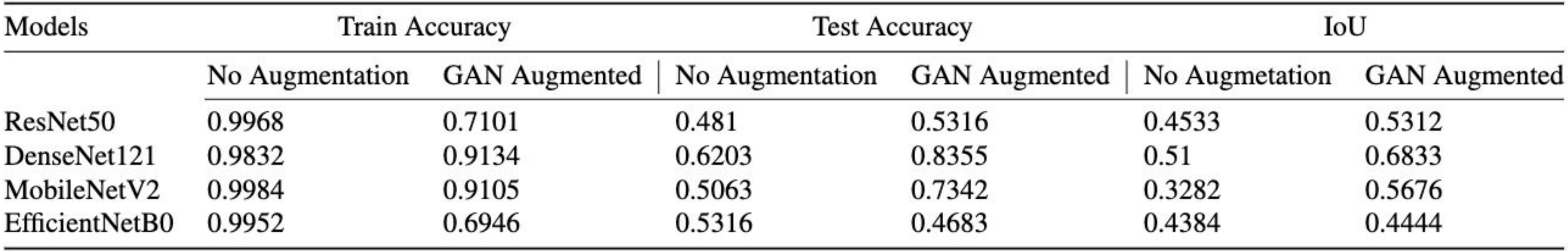
Conclusion

1. GAN-based augmentation improves model robustness and IoU scores
2. DenseNet121 and MobileNetV2 show good performance improvement with augmented data
3. Stable GAN training achieved with balanced generator/discriminator dynamics



Prototypical Network with ResNet50 Backbone N - Shots (2 way, 5 query) N	Training Accuracy (Best)	
	No Augmentation	GAN Augmentation (N/2)
10	0.5341	0.5127
20	0.601	0.5849
40	0.7327	0.7428

Table 2. Showing best accuracies while training using FSL algorithm and N. Prototypical Network has been used with ResNet50 backbone. Two classes being Malignant and Benign.



Models	Train Accuracy		Test Accuracy		IoU	
	No Augmentation	GAN Augmented	No Augmentation	GAN Augmented	No Augmetation	GAN Augmented
ResNet50	0.9968	0.7101	0.481	0.5316	0.4533	0.5312
DenseNet121	0.9832	0.9134	0.6203	0.8355	0.51	0.6833
MobileNetV2	0.9984	0.9105	0.5063	0.7342	0.3282	0.5676
EfficientNetB0	0.9952	0.6946	0.5316	0.4683	0.4384	0.4444

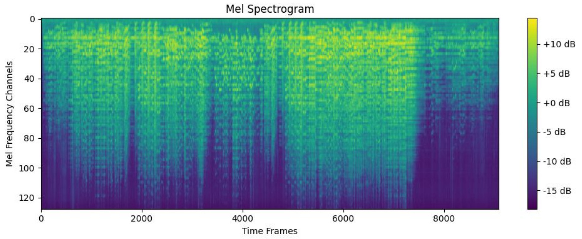
Table 1. Showing the different results obtained for Deep Learning models with and without augmentation using our proposed Augmentation Method

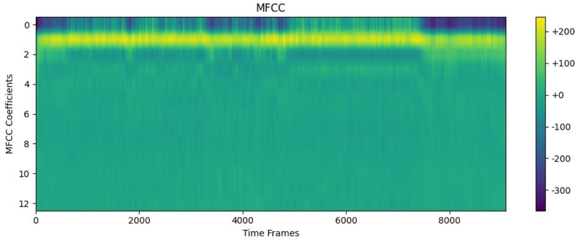
Audio to MIDI Conversion

Shreyan Mallik (smallik@umass.edu),
Shriram Giridhara (sgiridhara@umass.edu)

Project #78

Summary and Motivation





Problem Statement

Given a raw audio file, convert it to MIDI – a symbolic representation that encodes note sequences, pitch, and onset. Current methods often struggle with polyphonic music, where multiple notes (possibly from different instruments) overlap and interact.

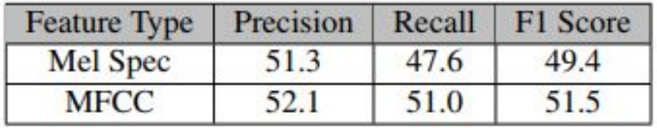
Project Goal

Train a deep learning model that effectively transcribes polyphonic audio, with an intention to generalize well for multiple instruments.

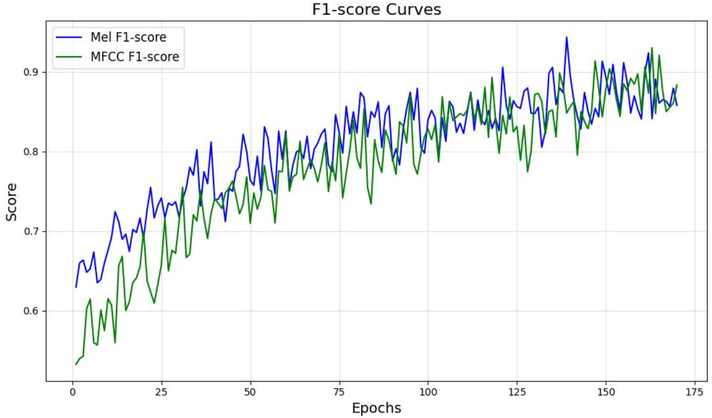
Approach

- Compose custom dataset containing audios from various instruments (piano, guitar, and drum)
- Test different audio transforms (Mel Spectrogram and MFCC) and compare effects on resulting MIDI using a CNN model

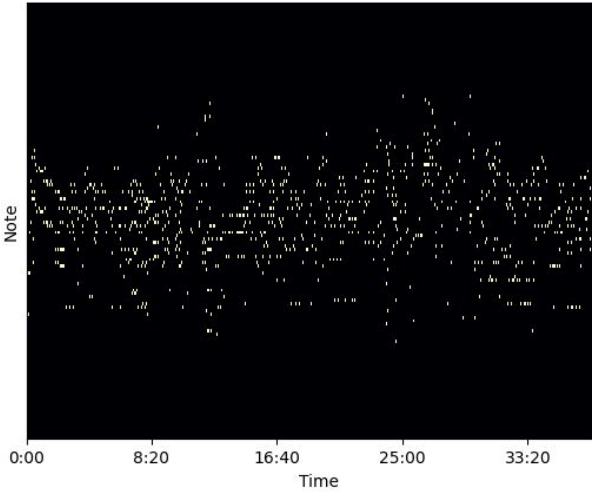
Results and Performance



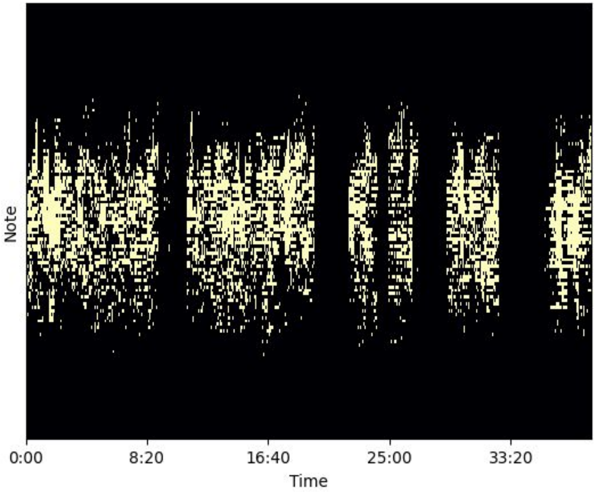
Feature Type	Precision	Recall	F1 Score
Mel Spec	51.3	47.6	49.4
MFCC	52.1	51.0	51.5



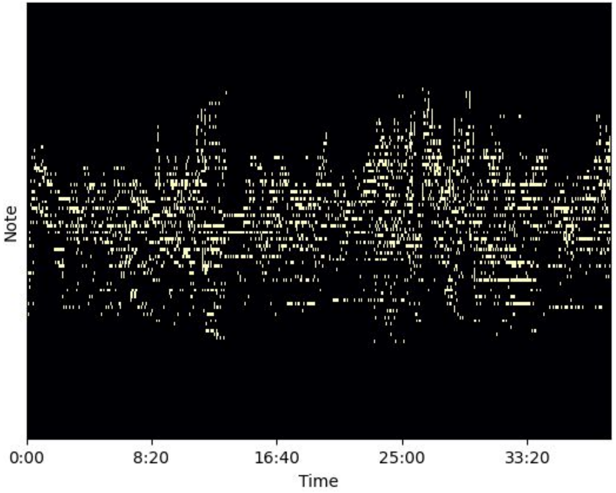
Actual MIDI



Mel Spectrogram - Prediction



MFCC - Prediction



Conclusion

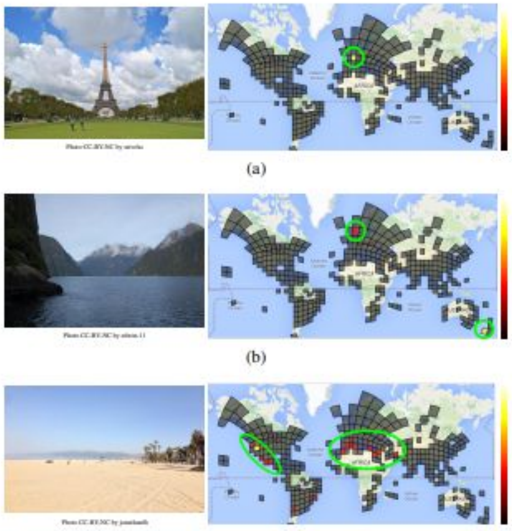
- **Datasets Used**: **MAESTRO**: Piano music. , **GuitarSet**: Guitar music, **Groove MIDI**: Drum music.
- **Feature Extraction Methods**:
  - **Mel Spectrogram**: Time-frequency representation based on human perception of pitch.
  - **MFCC (Mel-Frequency Cepstral Coefficients)**: Compressed and noise-resistant audio representation.
- **Model Architecture**:
  - Convolutional layers for low-level feature extraction.
  - Bi-LSTM for capturing temporal dependencies.
  - Fully connected layers for final MIDI output.
- **Evaluation**:
  - Compared performance using precision, recall, and F1 score.
  - Results show **MFCC** outperforms **Mel Spectrogram** in terms of precision, recall, and F1 score.
- **Key Insights**:
  - The model struggles with overlapping notes and complex instrument combinations.
  - MFCC has the potential to provide a more reliable representation for polyphonic audio transcription.
- **Future Work**:
  - Improve handling of overlapping instruments.
  - Incorporate more instruments and genres to enhance performance in general music transcription tasks, which could hint at future with live music transcription

Road signs for street recognition using CNNs

Algis Petlin (apetlin@umass.edu),
Hector Tierno (htierno@umass.edu),
Vinh Le (vinhle@umass.edu)

Project #79

Summary, motivation and background



(a)

(b)

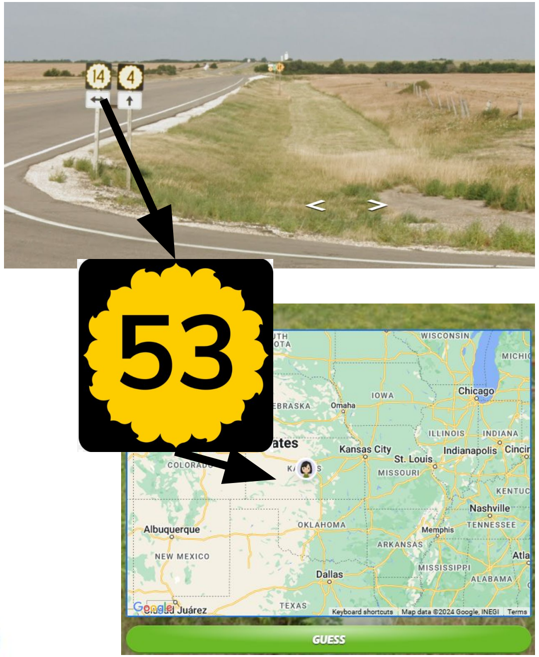
(c)

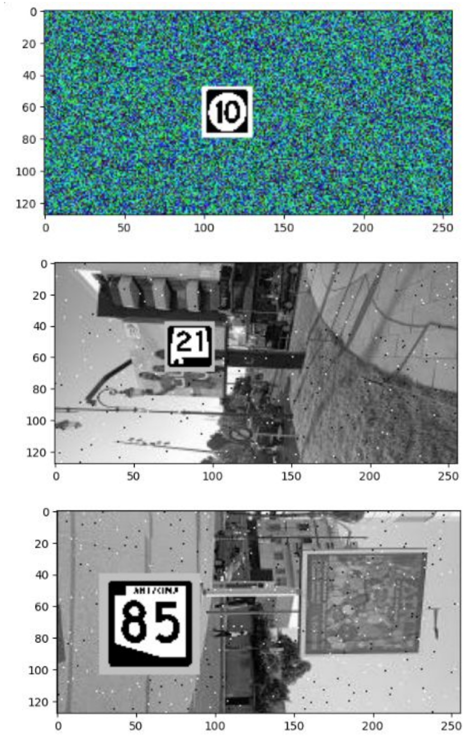
PlaNet - Photo Geolocation with Convolutional Neural Networks

Tobias Weyand
Google
weyand@google.com

Ilya Kostrikov
RWTH Aachen University
ilya.kostrikov@rwth-aachen.de

James Philbin
Google
philbinj@gmail.com





Main results

- Created 3 new datasets and a coding framework to continue to randomly generate new examples as needed.
- Tested our model trained with different training sets to compare disappointing, but hopefully still informative results.

Data Type	Val. Accuracy	Testing Accuracy
Random Background	~2%	–
‘Realistic’ Background	4-6%	2%
‘Realistic’ Background with Transformations	~3%	–

Conclusion and continuations

- Can this approach be comparable to object-detection with more data and a more complex architecture/pipeline?
- What is the best background for this approach? (what if we had used completely random images as background instead of ‘realistic’ ones?)

Video compression for panned camera videos

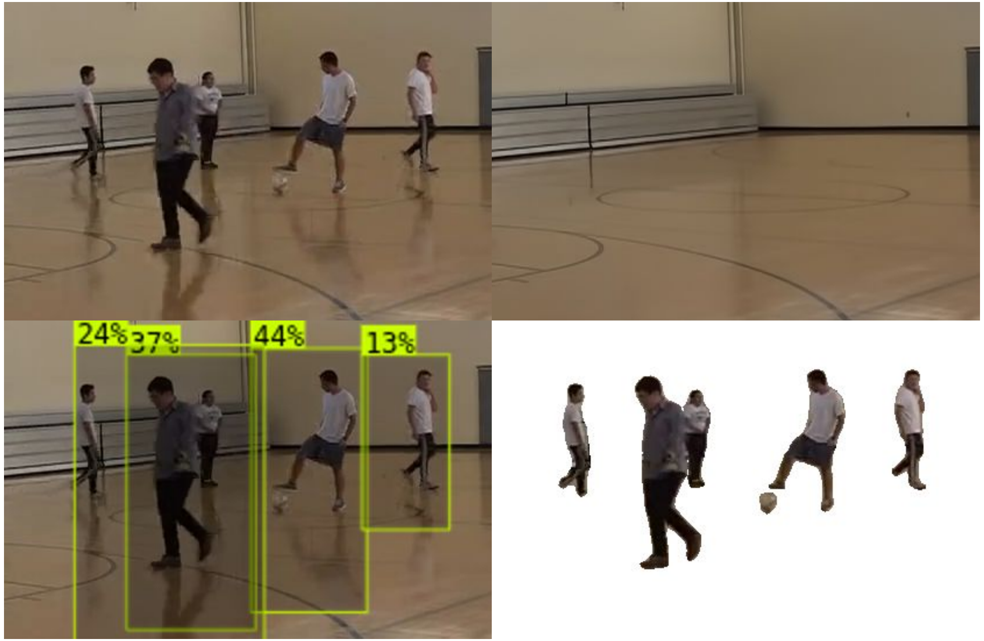
Anshul Vemulapalli (avemulapalli@umass.edu),
Eric Engelhart (eengelhart@umass.edu)

Project #84

Video Compression for Static and Panning Cameras

- Modern video codecs use I-frames (full images) and predicted frames (P-frames, B-frames) to compress video into reasonable sizes and maintain quality
    - When the background is repeated across the whole video, the I-frames may not be necessary
- Therefore, for this specific scenario, can we improve it?
- How:
    - Separate background from foreground.
    - Save foreground as video with solid background (run-length encoding means very little extra storage for I-frames)
    - Stitch background frames to create single background panorama, save crop coordinates for each frame

Results



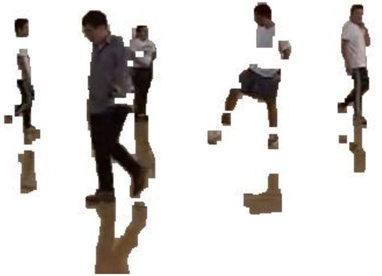
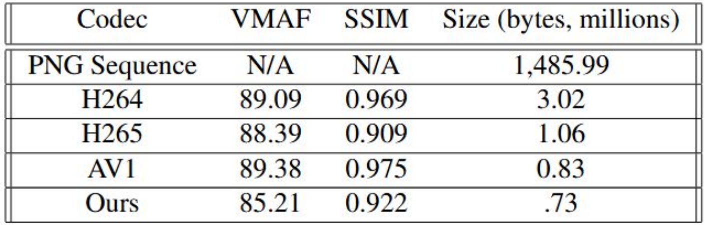


Table 2. SEPE[2] Clip 017 (Highway) 10 seconds



Codec	VMAF	SSIM	Size (bytes, millions)
PNG Sequence	N/A	N/A	1,485.99
H264	89.09	0.969	3.02
H265	88.39	0.909	1.06
AV1	89.38	0.975	0.83
Ours	85.21	0.922	.73

Conclusion

- Modern video codecs are incredibly good
  - But can still be improved upon for specific settings

- Lots of room for innovation
  - Last works we could find with “mosaic based video compression” are from ~2000!

- Applications:
  - Sports
  - Security footage
  - Potentially any footage from PTZ (pan, zoom, tilt) cameras

Scene Representations for Lifelong Embodied Exploration

Yuncong Yang (yuncongyang@umass.edu),
Zeyuan Yang (zeyuanyang@umass.edu)

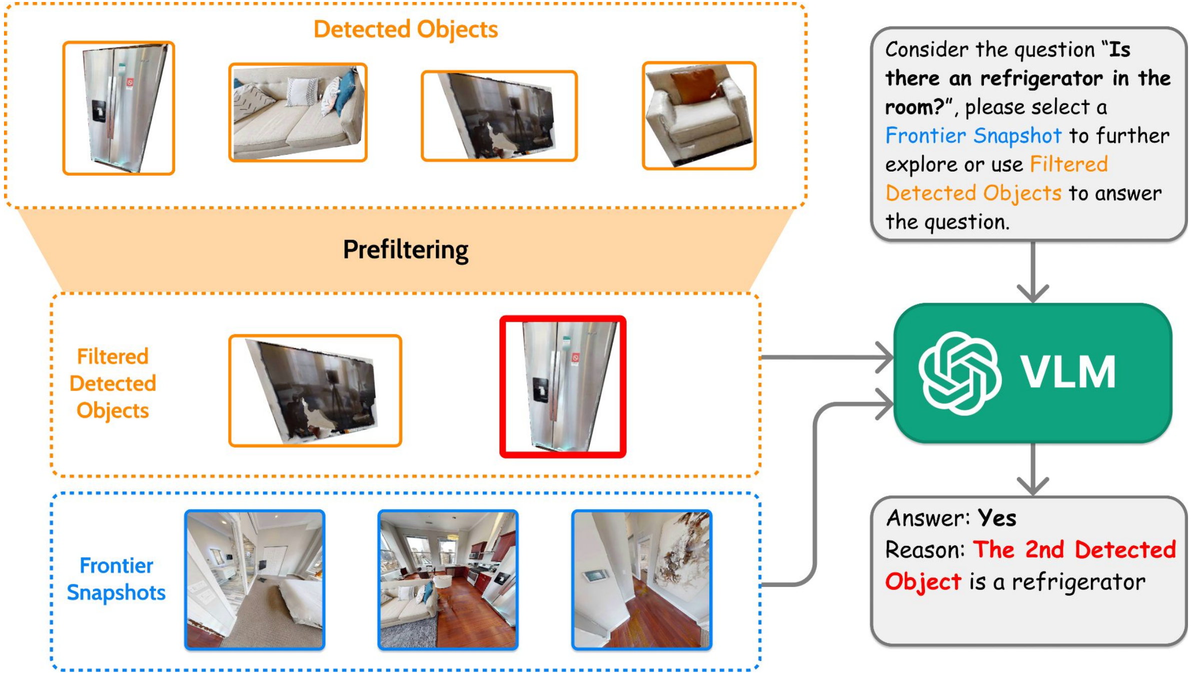
Project #87

Motivation

- In Embodied Exploration, not only representing the explored area is important. We need to represent the **unexplored regions** as well!
- During Embodied Exploration, the **scene memory grows** during the exploration. We need **memory retrieval** mechanism for lifelong exploration!

Technical Approach

- Detected Objects
  - Explored Regions
- Frontier Snapshots
  - Unexplored Regions
- VLM Reasonings
  - Decision Making

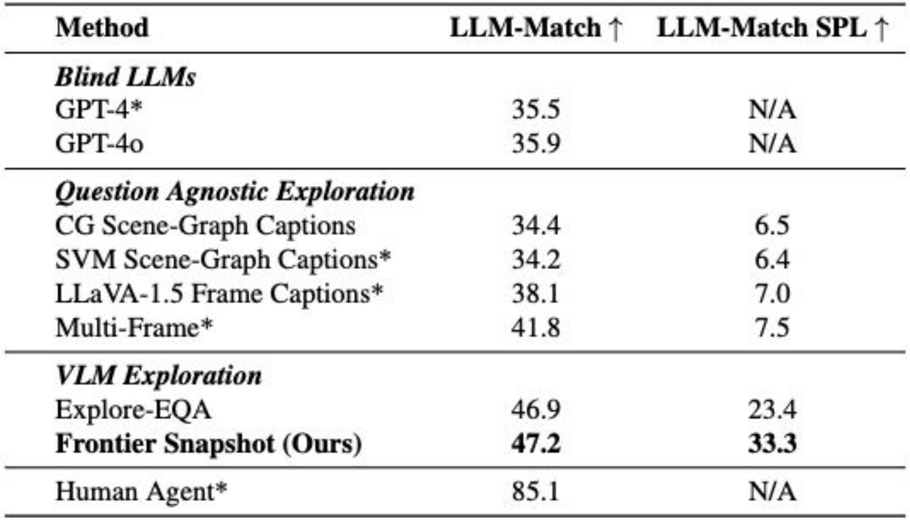


Consider the question "**Is there an refrigerator in the room?**", please select a Frontier Snapshot to further explore or use Filtered Detected Objects to answer the question.

Answer: **Yes**
Reason: **The 2nd Detected Object** is a refrigerator

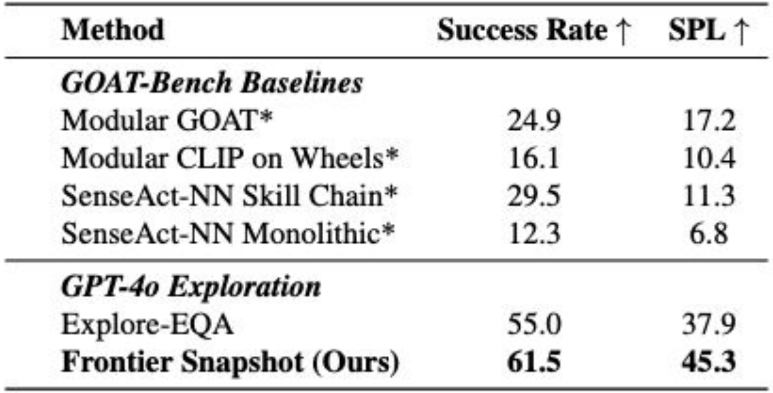
Experimental Results

- Superior performance across both benchmarks



Method	LLM-Match ↑	LLM-Match SPL ↑
Blind LLMs		
GPT-4*	35.5	N/A
GPT-4o	35.9	N/A
Question Agnostic Exploration		
CG Scene-Graph Captions	34.4	6.5
SVM Scene-Graph Captions*	34.2	6.4
LLaVA-1.5 Frame Captions*	38.1	7.0
Multi-Frame*	41.8	7.5
VLM Exploration		
Explore-EQA	46.9	23.4
Frontier Snapshot (Ours)	**47.2**	**33.3**
Human Agent*	85.1	N/A

Results on A-VQA



Method	Success Rate ↑	SPL ↑
GOAT-Bench Baselines		
Modular GOAT*	24.9	17.2
Modular CLIP on Wheels*	16.1	10.4
SenseAct-NN Skill Chain*	29.5	11.3
SenseAct-NN Monolithic*	12.3	6.8
GPT-4o Exploration		
Explore-EQA	55.0	37.9
Frontier Snapshot (Ours)	**61.5**	**45.3**

Results on GOAT-Bench

Great presentations, everyone! Thank you.

I hope this class gave you a sense of what deep learning can do and how to apply it effectively to specific applications. *It is not a silver bullet.*

There's still so much more to explore—topics like CV, NLP, and RL await you in the coming semesters. Attend seminars, faculty talks, and other events to learn more.

But first, take a well-deserved break!

And don't forget to complete the SRTIs. Your feedback is valuable!