REPORT DOCUMENTATION PAGE	Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
31-07-2022	Final Report	29-Jul-2021 - 29-Jul-2022

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Final Report: Third Wave Deep Learning Methods For Physical RealisRc Data	W911NF-21-1-0315
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHORS	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES	8. PERFORMING ORGANIZATION REPORT NUMBER
BlueLightAI, Inc 998 Cottrell Way Stanford, CA 94305 -1111	

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211	ARO
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)
	78875-CH-DRP.1

12. DISTRIBUTION AVAILIBILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Gunnar Carlsson
UU	UU	UU	UU		19b. TELEPHONE NUMBER
					650-575-8263

Standard Form 298 (Rev 8/98)
Prescribed by ANSI Std. Z39.18

RPPR Final Report
as of 02-Aug-2022

Agency Code: 21XD

Proposal Number: 78875CHDRP **Agreement Number: W911NF-21-1-0315**

INVESTIGATOR(S):

Name: Gunnar Carlsson
Email: gunnar@gunnarcarlsson.com
Phone Number: 6505758263
Principal: Y

Organization: **BlueLightAI, Inc**
Address: 998 Cottrell Way, Stanford, CA 943051111
Country: USA
DUNS Number: 117685174 EIN: 853111665

Report Date: 29-Aug-2022 Date Received: 31-Jul-2022
Final Report for Period Beginning 29-Jul-2021 and Ending 29-Jul-2022
Title: Third Wave Deep Learning Methods For Physical RealisRc Data
Begin Performance Period: 29-Jul-2021 **End Performance Period:** 29-Jul-2022
Report Term: 0-Other
Submitted By: Benjamin Mann Email: benjamin.mann@bluelightai.com
Phone: (703) 597-3237
Distribution Statement: 1-Approved for public release; distribution is unlimited.

STEM Degrees: **STEM Participants:**

Major Goals: This seedling will directly address DARPA’s Third Wave AI Initiative. BlueLightAI, Inc will build novel Deep Learning tools that contain contextual knowledge associated to a modality, incorporate prior knowledge through geometry, learn faster and more robustly, generalize better, be less susceptible to adversarial attack, give better compression of both data and algorithms thus allowing for sparser computation, will be ideally suited for edge computing, and will be equipped with diagnostics capabilities that permits analysis and provides transparency.

This seedling addresses many of the current limitations of AI and Deep Learning. These include requiring huge amounts of training data, which is not well suited to edge computing and to small sample sizes, failing to generalize well across similar dataset types, lacking transparency and diagnostic tools, being vulnerable to adversarial attacks, addressing only a limited number of cognitive tasks, and where human/machine interactions are limited.

Success will enable Deep Learning to improve human/machine interactions, generalization and robustness, transparency and diagnostics, operate better on smaller datasets, and support wider ranges of cognitive tasks. In particular, success will offer a paradigm shift from training on big data for small insights (those that generalize poorly) to training on small data for bigger insights (those that generalize better).

Accomplishments: The project was completed on schedule with all tasks completed and all deliverables presented to ARO and DARPA. Monthly reports give a complete picture of the dynamics of our work and progress.

The project accomplished the main goals of the seedling.

The final report, uploaded below under the "upload" button, contains a complete and detailed report of results, accomplishments, lessons learned, and directions for further work.

Training Opportunities: Nothing to Report

Results Dissemination: Nothing to Report

Honors and Awards: Nothing to Report

Protocol Activity Status:

Technology Transfer: Nothing to Report

PARTICIPANTS:

Participant Type: PD/PI
Participant: Gunnar Carlsson
Person Months Worked: 12.00 **Funding Support:**
Project Contribution:
National Academy Member: N

Participant Type: Co PD/PI
Participant: Benjamin Mann
Person Months Worked: 12.00 **Funding Support:**
Project Contribution:
National Academy Member: N

Participant Type: Co-Investigator
Participant: Nikhil Kotecha
Person Months Worked: 10.00 **Funding Support:**
Project Contribution:
National Academy Member: N

Partners

,

I certify that the information in the report is complete and accurate:
Signature: Benjamin Mann
Signature Date: 7/31/22 12:59PM

BlueLightAI, Inc Final Technical Report for July 2022
DARPA Seedling/ARO Contracting
"Third Wave Deep Learning Methods For Physical Realistic Data "

- Contract Number: W911NF2110315.
- DARPA Program Manager: Dr. Anne Fischer.
- ARO Contracts Officer: Mr. Barry Covert.
- ARO COR: Dr. Dawanne Poree.
- Deliverable Number: 12.
- CLIN 0001 AM.
- Invoice amount: $25,136.45.
- Date submitted: July 31, 2022.
- Submitted by BlueLight AI, Inc.
- For any questions regarding this report please contact Dr. Benjamin Mann, Co-Founder and VP of operations, benjamin.mann@bluelightai.com, 703-597-3237.

This final technical report is the twelfth and final monthly deliverable on this contract. Per the contract, an annual (through 12/31/2021) and final (6/29/2022) financial report will be submitted separately at the appropriate times. Since this is a firm, fixed price, grant we will also include, on a monthly basis here, our history of submitting invoices to WAWF.

All references to the work below are contained in the SOW in our original proposal. In particular, the schedule of tasks is summarized in the following Gantt chart taken from our proposal.

Schedule for Tasks for BlueLightAI DARPA seedling proposal

TASK	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Month 7	Month 8	Month 9	Month 10	Month 11	Month 12
1	X											
2	X	X	X									
3				X								
4					X							
5						X						
6							X	X	X			
7										X	X	
8												X

Executive Summary

The main body of this final report consists of seven sections that detail the purpose of the effort, the main problem that we addressed, the main technical results not covered in our previous monthly reports. the main take-aways, lessons learned (what worked and what didn't), and future direction to explore. We conclude this report with two final sections on a summary of previously completed tasks and financial reporting.

Purpose

The purpose of this effort was to directly address DARPA's Third Wave AI Initiative by building novel Deep Learning tools that contain contextual knowledge associated to a modality, incorporate prior knowledge through geometry, learn faster and more robustly, generalize better, be less susceptible to adversarial attack, give better compression of both data and algorithms thus allowing for sparser computation, will be ideally suited for edge computing, and will be equipped with diagnostics capabilities that permits analysis and provides transparency.

In particular, we sought to produce a proof-of-concept that addressed many of the current limitations of AI and Deep Learning. These include requiring huge amounts of training data, which is not well suited to edge computing and to small sample sizes, failing to generalize well across similar dataset types, lacking transparency and diagnostic tools, being vulnerable to adversarial attacks, addressing only a limited number of cognitive tasks, and where human/machine interactions are limited.

The ultimate goal of this effort is to provide the first steps toward "augmented" Deep Learning tools to improve human/machine interactions, generalization and robustness, transparency and diagnostics, operate better on smaller datasets, and support wider ranges of cognitive tasks. In particular, success will offer a paradigm shift from training on big data for small insights (those that generalize poorly) to training on small data for bigger insights (those that generalize better).

The Main Problem

In this seedling effort we proposed to adapt our novel BlueLightAI deep learning algorithms to build novel Topological Neural Networks (TNNs) to see if they would give better results than the current Convolutional Neural Networks (CNNS) on dynamic echocardiogram data. The test data set, EchoNet, was a well-known and studied dataset consisting of video clips of echocardiogram readings from the Stanford Medical Center. This dataset contained the attributes (noisy and dynamic) on which we wanted to test our methods. Furthermore, the group at Stanford had done an extensive analysis on the dataset using traditional CNN methodologies. For those reasons, we felt that this dataset was an ideal test case to use in this seedling. That did, in fact, turn out to be the case. At the end of this report the reader is referred to the list of tasks from our SOW, which gives the road map we followed in designing, implementing, and testing our TNN architectures as well as benchmark comparisons with more traditional approaches.

Summary of Results not Covered in Previous Monthly Reports

The goal for this month was to determine the robustness of our findings, and implement minor upgrades for incremental improvement. To validate the robustness of the Klein Tangent feature TNNs, also known as Tangent Bundle Neural Networks (TBNNs) results we performed multiple runs of the models on both synthetic and real data. The purpose of the synthetic data was to take an extreme condition of the data - e.g. outliers or data with high degree of noise - and verify the results of the models without biasing the underlying data sets. For instance, we explored the degradation of the models under different artificially injected noise distributions to ensure the model was learning robust features and not spurious indicators, such as image quality reflecting the age of the ultrasound devices. (These robustness tests performed largely fall under the umbrella of adversarial attacks, with the previously described injection of noise called data poisoning.) Further, we modified the TBNNs to include a version 1 and version 2. TBNN version 1 contains a "vanilla" architecture without the empirical tricks applied from the deep learning literature. TBNN v2 contains the same architecture with deep learning modifications that have been shown to improve performance. Modifications include: batch normalization, He initialization, and dropout. (Note: The deep learning modifications were all applied to the baselines and the models with highest performance were selected.) Additionally, models were run 10 times with results averaged across the runs. The architecture and data between runs were untouched to prevent introduction of bias.

For the TBNN v1 and TBNN v2 results marked an increased in all core performance metrics: model performance as measured by Mean Absolute Error (MAE), model training time, and data efficiency. TBNN v1's performance notes are: 6.1 +- 0.6 MAE and a model training time of 331 minutes. TBNN v2's performance notes are: 5.6 +- 0.5 and a model training time of 311 minutes. Results and comparison of results to baselines are detailed in sections below.

As promised, as one of the final deliverables, a final status report for this seedling effort was presented via a webinar on July 27, 2022 to stakeholders. This final report represents the other final deliverable. It is being submitted through the ARO web portal https://extranet.aro.army.mil. In addition, the slides used in the final webinar presentation are appended to this report for completeness.

Main Take-Aways

1. Why try TNNs and TBNNs?

CNNs are powerful tools that yield high performance in classification and regression for a variety of tasks when trained on large amounts of data. Examining the cause of the decision produced by a CNN can be difficult because the algorithm is designed to optimize for performance and without consideration of intrinsic explainability. Post hoc analysis interpretation methods exist and can be useful to visualize and understand the weights associated with certain decision elements in trained CNNs, but these methods leave much to be desired. Instead TNNs (topological neural networks) are designed with interpretability baked into the intrinsic structure of the models. Before exploring specific mechanisms to interpret the process and results of decision making in TNNs versus CNNs, we wanted to ensure other core metrics, such as performance, speed, and data efficiency were not sacrificed.

2. Construction of TNNs and benchmark comparisons with CNNs

To effectively measure the performance of TNNs, baselines needed to be established. Baselines were established for models trained on static images and for models trained on videos. After selecting the EchoNet dataset, we replicated the results published in the accompanying paper[1]. Our results reflected similar numbers presented in the paper within an acceptable level of deviation.

Table 3: Benchmark Model Performance on Test Set (Hyperparameters chosen from Validation Set)

Task	Model	Clip Length	Sampling Rate	MAE	RMSE	R^2
Ejection Fraction	Human Experts	Entire Video	Every Frame	3.12	4.57	0.88
Ejection Fraction	R3D	16	1 in 4	5.44	6.16	0.71
Ejection Fraction	MC3	16	1 in 4	5.91	6.80	0.69
Ejection Fraction	R2+1D	16	1 in 4	6.87	7.55	0.66
End Systolic Volume	R3D	16	1 in 4	12.7	19.3	0.72
End Systolic Volume	MC3	16	1 in 4	12.4	18.3	0.71
End Systolic Volume	R2+1D	16	1 in 4	12.4	19.7	0.74
End Diastolic Volume	R3D	16	1 in 4	20.0	30.3	0.64
End Diastolic Volume	MC3	16	1 in 4	51.8	35.2	0.61
End Diastolic Volume	R2+1D	16	1 in 4	21.1	28.8	0.60

To compare results of TNNs, we coded a drop in replacement for CNN layers with Klein Parameterized layers with a set of filters constrained to the Klein bottle. We used the same resnet-50 baseline architecture but with replaced TNN layers, and used the same compute environment. The preliminary results were promising: faster training times of 3 days to 2.25 days, results with 18% less data, and worse, but close results. NOL refers to the standard CNN model. NOL+ refers to a modified architecture with improvements. KF refers to the Klein Features model, also known as TNNs.

Effort was then focused on Tangent Bundle Neural Networks (TBNNs). TBNNs computed 3d convolutions with a set of filters constrained to lie on the tangent bundle of the translational Klein bottle. The model was programmed to serve as a drop in replacement for the standard PyTorch 3D convolution models. To effectively evaluate the performance of TBNNs, we implemented a suite of 2D and 3D CNNs. The highest performers are presented in the following table and graph.

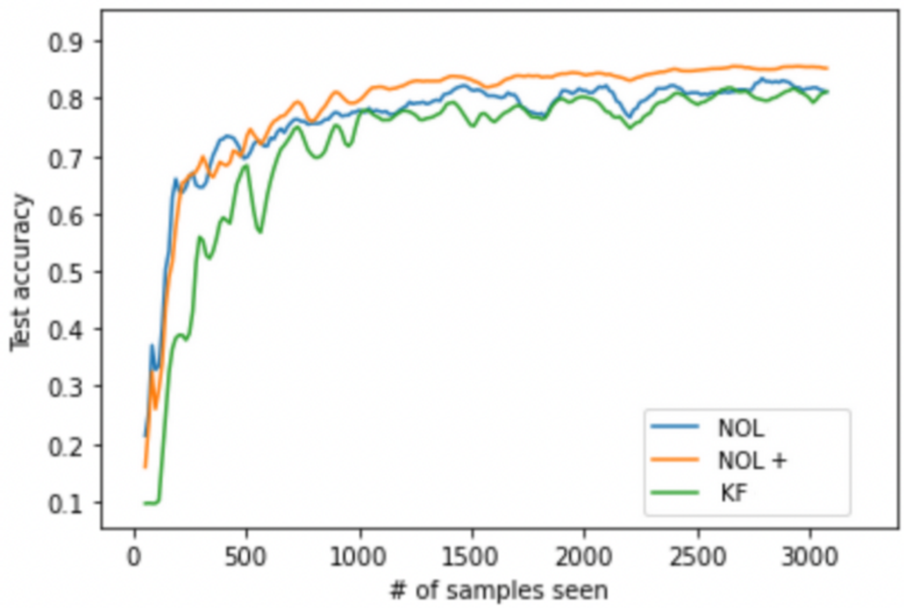
Please note:

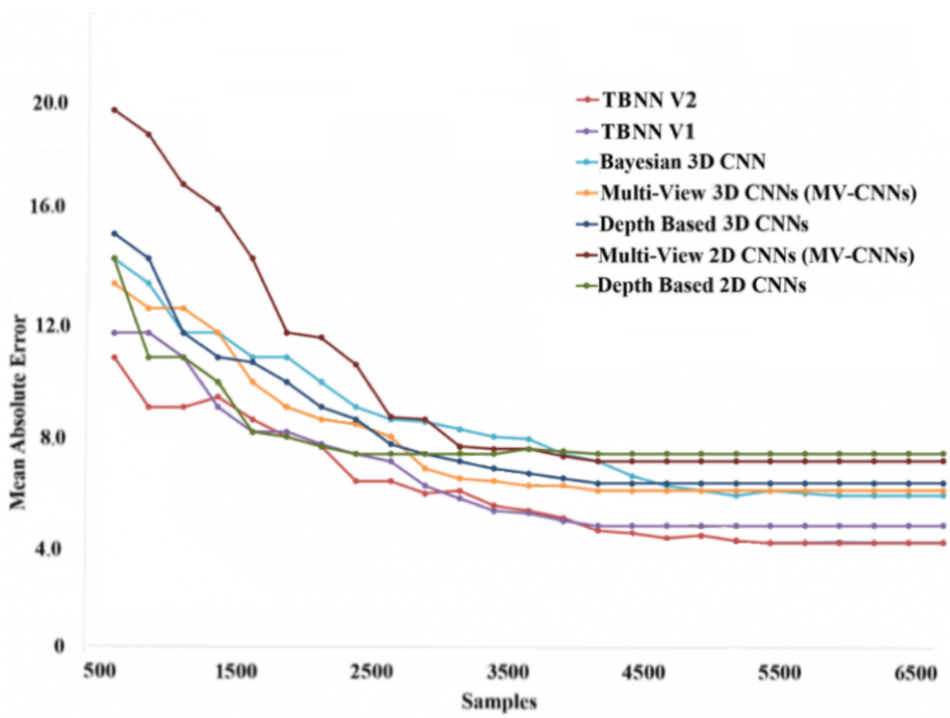
1. In the web debrief, there was specific mention of including the results from the original paper with the results from this seedling effort. Rows 2, 4, and 7 are from the original work.

2. Results are ordered in order of highest Mean Absolute Error (MAE). Results will be interpreted in the section below.

---

[1] Ouyang, David, et al. "Echonet-dynamic: a large new cardiac motion video data resource for medical machine learning." *NeurIPS ML4H Workshop: Vancouver, BC, Canada*. 2019.


0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
Test accuracy
0
500
1000
1500
2000
2500
3000
# of samples seen
NOL
NOL +
KF


20.0
16.0
12.0
8.0
4.0
0
Mean Absolute Error
500
1500
2500
3500
4500
5500
6500
Samples
TBNN V2
TBNN V1
Bayesian 3D CNN
Multi-View 3D CNNs (MV-CNNs)
Depth Based 3D CNNs
Multi-View 2D CNNs (MV-CNNs)
Depth Based 2D CNNs

Task	Model	Mean Absolute Error (MAE)	MAE standard deviation	Model Training Time (minutes)	Samples to Convergence (n=6500, round numbers), “asymptotic relative efficiency”	Samples for Top model to reach converged max performance of comparison model	Data Efficiency relative to top model
Ejection Fraction (EF)	Human Experts	3.12	NA	NA	NA	NA	NA
EF	R3D (Nature Model trained on clips)	5.44	NA	NA	NA	NA	NA

EF	TBNN V2	5.6	+- 0.5	311	5750 samples to converge (at least 2 iterations of stability)	5750	5750/5750 = 100%
EF	MC3 (Nature Model trained on clips)	5.91	NA	NA	NA	NA	NA
EF	TBNN V1	6.1	+- 0.6	331	4750	4250	4250/4750 = 89%
EF	R2+1D (Nature Model trained on clips)	6.87	NA	NA	NA	NA	NA
EF	Bayesian 3D CNN	7.0	+- 0.9	424	6250	3000	3000/6250 = 48%
EF	MV CNN 3D	7.2	+- 0.8	387	4750	2500	2500/4750 = 53%
EF	DB CNN 3D	7.4	+- 0.6	365	4750	2250	2250/4750 = 47%
EF	MV CNN 2D	8.3	+- 0.4	321	5000	2000	2000/5000 = 40%
EF	DB CNN 2D	8.8	+- 0.5	248	4500	2000	2000 / 4500 = 44%

Lessons Learned: What Worked

1. Coding, running, testing all worked

In the software development process, there was attention paid to ease of use and reusability to replicate experiments, and to extend the code base to new problem domains. The code was thus designed to follow the design patterns of PyTorch, with TNNs and TBNNs to serve as drop-in replacements for 2D and 3D CNNs. Standard software development practices were followed to ensure the code was performing in expected ways with predictable failure modes. For instance, smoke tests, negative testing, etc.

2. Summarize improvements over benchmarking earlier results

Results from the above graphs and tables indicate the utility of the TBNNs. The TBNNs were as performant as top models in the benchmark paper. The models from the benchmark paper were trained on clips. To perform a more equitable comparison for models trained on video, we also ran a derby of 3D and 2D CNNs and selected the highest performers. From the other performance metrics of model training time and data efficiency relative to the other models the TBNNs outperform the baselines resulting in faster, more data efficient models.

3. Evidence to continue research

The core hypothesis that high performance models can be designed to be structurally interpretable and efficient in terms of training time and usage of data has been sufficiently validated for further research. Interpretability and explainability of results both need additional evidence.

Lessons Learned: What Didn't Work

1. What limitations do we observe?

    A. It is not yet clear under what conditions the models are effective. For instance, sufficient evidence has not been gathered to determine the performance of TNNs and TBNNs in different conditions - e.g. smaller data slices, noise injected data, robustness under adversarial attack, etc.

    B. It remains unclear the optimal nature of hybrid architectures (CNNs and TNNs and TBNNs) and the optimal conditions for hybrid architectures. Open conditions include what is the appropriate distribution for CNNs and TNNs and TBNNs. Currently TNNs and TBNNs are "pure" models - i.e. with no CNN modules. The interacting effects would be worthwhile to explore.

    C. Learning dynamics for TNNs were initially inferior to the CNN benchmarks. We were unable to replicate this inferiority in TBNNs.

2. What features / architectures did we try that didn't seem to be useful.

A. The TBNN v2 model served as our test of whether traditional deep learning upgrades add value to the vanilla TBNN v1. Examples of upgrades include: batch normalization, He initialization, and dropout. These results provided marginal benefit in performance, speed, and data efficiency. It is unclear the robustness of these results and the impacts to interpretability. Without sufficient evidence to validate the claim either way, we prefer to put this in the didn't work section.

3. What do we regret not trying?

   A. We regret not trying visualization methods to interpret the learning process of the TNNs and TBNNs. The structure of the TNNs and TBNNs enables interrogation of the weights and dynamics of the learning process, and not trying these methods is suboptimal.

   B. Visualization of results

      i. Visualizing and interpreting the results directly onto representations of the heart would have been a powerful example to take to cardiologists to validate the learning process of the TBNNs and TNNs. The software behind a visualization suite was sketched out in block diagrams, but did not get implemented due to time constraints.

   C. Exploring hybrid models

      i. We alluded to this open question previously: What are good TNN and TBNN problems? How can TNNs and DNNs complement each other? Understanding under what conditions TNNs and TBNNs in isolation and in combination with CNNs and other Deep Learning architectures are optimal would be useful. Specifically, we would like to understand the efficacy of TBNNs and TNNs on different kinds of data - e.g. text, audio, multimodal cases, time series, etc. - of variable quality, noise, and size. Further, the efficacy of TNNs and TBNNs in different learning modalities - e.g. active, few shot, real time, federated - is an important question to investigate.

Future Directions to Explore

1. Enlarging the "Klein bottle" features for video: In other efforts we have explored a wide variety of ways to enrich the "Klein bottle" features (including textures, different grid sizes, and enlargements to include bull-eyes and poke-a-dots as basic patches) in analyzing static images. It would be natural to extend those methods and results to video and build on the results we obtained in this effort.

2. Attention based methods: Attention based methods in Deep Learning have been extremely powerful in both text analysis and static image analysis. In future work, we would like to

study and implement how we could incorporate attention based methods into our current design; that is, extend attention based methods from static images to video clips.

3. Deeper interrogation of particular slice data, modifying conditions to understand robustness, and analyzing failure modes: By that, we specifically mean how to seek answers on the conditions with which TNN models and deep learning models provide superior results, and under what conditions traditional and TNN models can be combined to provide superior performance to the individual elements. Again, in a global sense, such understanding of the kinds of data, quality of data, etc, was certainly outside of scope of this seedling. However, it strikes us as potentially both natural and powerful to pursue in the future.

Summary of the Results to Date

Here is list of the tasks from the original proposal of the work. They have all been completed on schedule. All the work done to complete these tasks was described in detail in our monthly reports as well as two web meetings with the PM and COR. Therefore, we refer to those reports and slide decks, all of which were provided at the time to both government officers, and are therefore part of the reporting record, for detailed descriptions of our month-by-month progress.

Current Final Task Completed This Month

Task 8: (month 12) The project culminates in preparing the two main deliverables. That is, we will polish our final constructions, verify our final results, and then submit a final report and present our findings to the DARPA PM and ARO COR.

List of Earlier Completed Tasks with Completion Dates

This is the list of tasks taken from the proposal. All tasks were completed on schedule.

Task 1: (month 1) Prepare echocardiogram data for ingestion into TNN. This is a purely data wrangling task that requires only well-known methods.

Completed August 29 2021.

Task 2: (months 1, 2, 3) Design and implement a first cut TNN for the learning components using the known constructions such as ResNet as well as our experience with previous TNN constructions, [4], [5], [11]. There will be several choices to make here for the hyper-parameters in this initial construction. To begin, we shall use the methods of [11], [12] on static slices of the training video clips to identify the smallest subset of the Klein bottle to use as the fundamental building block for our first proposed candidate for our Platonic model. Once such a subset is identified, it can be extended by translations and tangent bundle constructions to the Platonic model to be used in this task. Then we will experimentally explore selecting the hyper-parameters to build the individual primitives, the number of primitives in each layer, as well as the resulting modifications within the ResNet. This is a complicated task, which will be directed by the PI.

Completed November 29 2021.

Task 3: (month 4) Tune and train the TNN implementation in task 2 on the full training sample of the selected video clips. In particular, this will establish the weights needed to fix the internal algorithmic architecture of this TNN. The determination of these weights will be important in task 4 for testing and in task 5 for diagnostic purposes.

Completed January 29 2022.

Task 4: (month 5) Test the TNN in task 3 on the complement of the training sample. It is at this point that we will compare our results to the benchmark results given in [18] to better judge the performance so far for our TNN.

Completed January 29 2022.

Task 5: (month 6) Using the results of tasks 2, 3, and 4 the team will deploy the proprietary BlueLightAI diagnostic tools [11], [12] on the architecture of the TNN implemented in task 3. This will enable us to better understand the relative strengths and weaknesses of the test results obtained in task 4.

Completed January 29 2022.

Please note in the original proposal we broke the work for Task 3, 4, and 5 out separately. However, when beginning to carry out the work, we realized that the tasks were best undertaken in a back and forth iterative way. Thus, we pursued that strategy of producing a better combined result by simultaneously working both. We completed this combined work this month using this unified approach. Taking together this work kept us on schedule as well as set us up well for the later tasks.

Task 6: (months 7-9) We will refine the original construction outlined in tasks 2-5 above. That is, based on the accuracy and the diagnostic results from the first iteration, we will refine the TNN constructed in task 2 by iterating our construction. This will involve both refining the choice of Platonic model as well as adjusting the composition and number of layers as well as the hyper-parameter choices for the primitives from task 2. In particular, we expect to add additional layers as well as enlarging the primitives to larger patch and grid size; that is, use components P(m,n,k,s) for values of m and n greater than three.

Task 7: (months 10-11) Repeat tasks 3, 4, and 5 where the TNN from task 2 has been replaced by the new TNN constructed in task 6.

Completed June 29 2022.

Task 8: (month 12) The project culminates in preparing the two main deliverables. That is, we will polish our final constructions, verify our final results, and then submit a final report and present our findings to the DARPA PM and ARO COR.

Completed July 29 2022.

Closing Technical Statement

- We have completed all proposed tasks, summarized our technical findings in this final technical report, and presented our findings to the DARPA PM and ARO COR.

Financial Report

- This is a firm-fixed price contract.
- There are 12 milestones/deliverables on this contract. Each deliverable is due at the end of the corresponding 12 months of performance as given in the contract.
- The previous eleven monthly reports constituted the first eleven deliverables. Invoices for these deliverables were submitted through WAWF in conjunction with each monthly report. Thus, this first eleven deliverables were submitted on time.
- This final technical report constitutes the twelfth and final deliverable. The contract states that the reimbursement for this last deliverable is $25,136.45. An invoice is being submitted through WAWF in conjunction with this final report. Thus, this final deliverable is being submitted on time.
- In a separate document we submitted the annual financial report for the period 7/29/2021-12/31/2021 in January 2022 to the COR. It contains the total amount invoiced and received in 2021 and is consistent with the reporting here.
- In a separate document we will submit our final financial report, after receiving our final payment. This will allow us to satisfactorily complete and thus close this contract.

BlueLightAI Status Report to DARPA/ARO on Seedling Grant: Third Wave Deep Learning Methods for Physical Realistic Data

Nikhil Kotecha, Gunnar Carlsson, Benjamin Mann
7/27/22

Executive Summary

- Presenting final report on how Klein Tangent Bundle Models perform.
- As hoped, new models outperform best in class, state of the art models.
  - Fast
  - Less Data
  - More performant on core metrics
- First section of presentation, recalls prior work.
- Second section of presentation, gives detailed report on results.
- Third section, discusses directions for further research.

Project Timeline

- Complex TNN
    - End of March
- Recent Past - task 7 (TBNNs)
    - Months of April, May
- Final Report
    - June

Prior work

- General Background
- Data intro
- Baseline
  - Architecture
  - Baseline output
- TNN background
- Complex TNN
  - Completed in March

General Background

- Echocardiography, or cardiac ultrasound, is the most widely used and readily available imaging modality to assess cardiac function and structure.
  - Portable instrumentation
  - Rapid image acquisition
  - High temporal resolution
  - Without the risks of ionizing radiation
  - Backbone of cardiovascular imaging.

Data Introduction

- Noisy, Complex, Dynamic Data
- Use EchoNet-Dynamic Database
  - 10,030 labeled echocardiogram videos
  - Human expert annotations (measurements, tracings of left ventricle, and calculations)
  - Baseline to study cardiac motion and chamber sizes.

Dataset Label Variables

Variable	Description
FileName	Hashed file name used to link videos, labels, and annotations
EF	Ejection fraction calculated from ESV and EDV
ESV	End systolic volume calculated by method of discs
EDV	End diastolic volume calculated by method of discs
Height	Video Height
Width	Video Width
FPS	Frames Per Second
NumFrames	Number of Frames in whole video
Split	Classification of train/validation/test sets used for benchmarking

Baseline

- ResNET-18
  - Base architecture and consists of 18 convolutional layers with residual connections connecting odd numbered layers
  - Three models were explored – focus on R2+1D
    - R2+1D factorizes 3D convolutional filters into separate spatial and temporal components with different spatial and temporal sizes such that total number of parameters is similar to filters fo equal size in dimension width, height and time.
    - Clip length of 16 and frame sampling rate of 4 performed best
    - MAE 6.87%
    - Human accuracy has been described at MAE of 4-5% for skilled echocardiographers in controlled settings

Baseline Results

- Results from Baseline

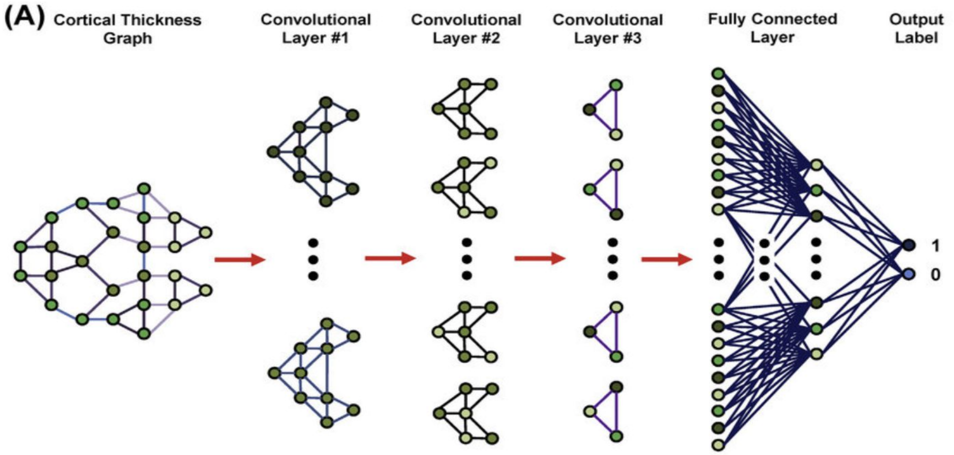
Table 3: Benchmark Model Performance on Test Set (Hyperparameters chosen from Validation Set)

Task	Model	Clip Length	Sampling Rate	MAE	RMSE	R^2
Ejection Fraction	Human Experts	Entire Video	Every Frame	3.12	4.57	0.88
Ejection Fraction	R3D	16	1 in 4	5.44	6.16	0.71
Ejection Fraction	MC3	16	1 in 4	5.91	6.80	0.69
Ejection Fraction	R2+1D	16	1 in 4	6.87	7.55	0.66
End Systolic Volume	R3D	16	1 in 4	12.7	19.3	0.72
End Systolic Volume	MC3	16	1 in 4	12.4	18.3	0.71
End Systolic Volume	R2+1D	16	1 in 4	12.4	19.7	0.74
End Diastolic Volume	R3D	16	1 in 4	20.0	30.3	0.64
End Diastolic Volume	MC3	16	1 in 4	51.8	35.2	0.61
End Diastolic Volume	R2+1D	16	1 in 4	21.1	28.8	0.60

Source: https://echonet.github.io/dynamic/NeuroIPS_2019_ML4H%20Workshop_Paper.pdf

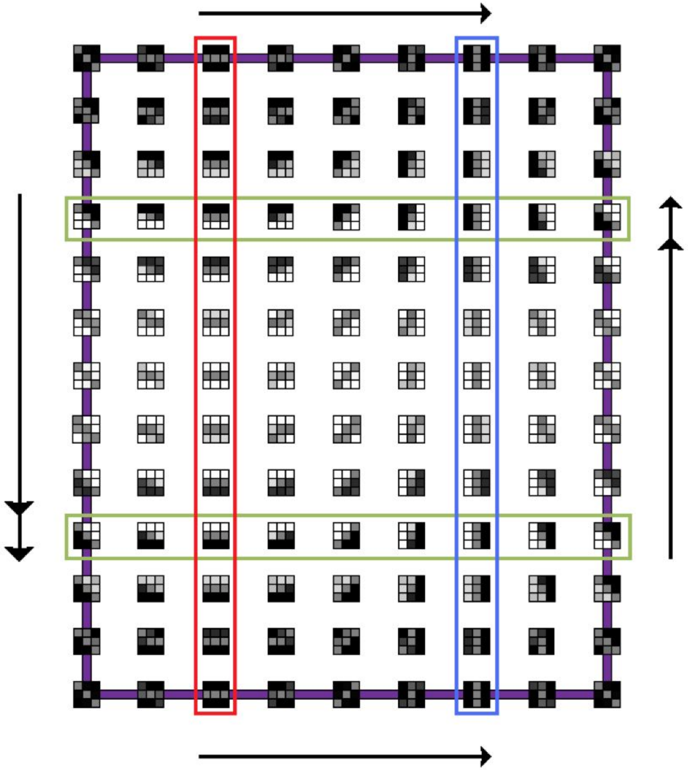
TNN Background - Reframing CNNs

- Topological structure on the feature space can be used to inform the structure of neural networks.
- When the set of features is a graph, one can construct a network layer with values of output features dependent only on values of input features for features in a neighborhood of the input.



Wee, Chong-Yaw, et al. "Cortical graph neural network for AD and MCI diagnosis and transfer learning across populations." *NeuroImage: Clinical* 23 (2019): 101929.

TNN Background - Klein Bottle Patches

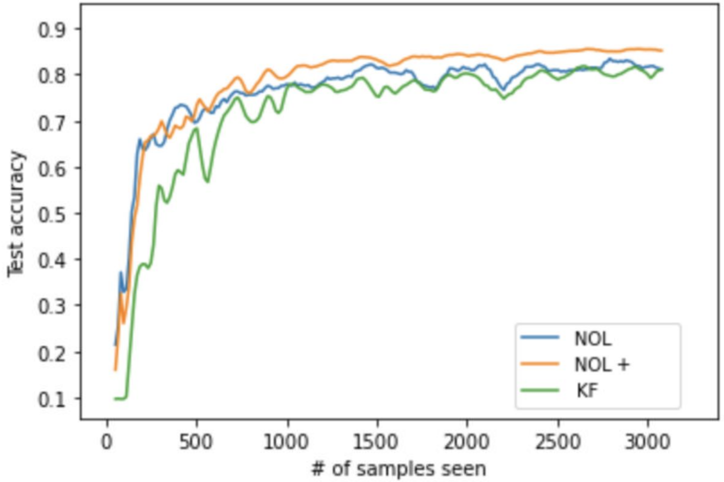


TNN Background

- For images, CNNs learn a collection of convolutional filters, which can be thought of as image patches.
  - This collection of filters can be interpreted.
- If CNN’s learn Klein bottle patches as convolutional filters, and if those patches are assigned as filters at initialization then one gets a “Topological CNN.”
- Implementation:
  - Convolutional kernels are constrained to the Klein bottle.
  - It has two parameters per kernel, corresponding to the two dimensions of the Klein Bottle.
  - Parameters are updated like any other parameter - i.e. in same fashion as standard PyTorch 2D CNN.

Complex TNN (through task 6, March 2022)

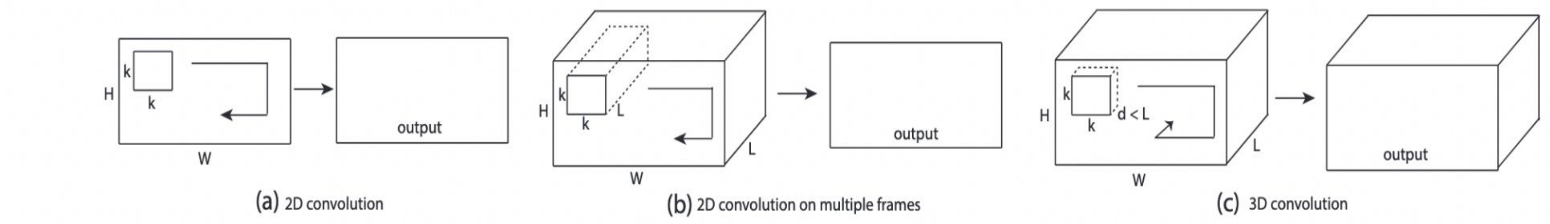
- Trying to find balance between CNN and TNN layers
  - Created drop in replacement for CNN layers with Klein Parameterized Layers in resnet 50 to create more complex, hybrid model
    - Exploring interaction effects between CNN and TNN layers
- Preliminary results are (not much) worse than deep learning
  - Faster training - 25% faster, 18% less data
  - Potential of interpretability
  - Still need to finalize architecture and tuning
  - After will calculate
    - Sample metrics
    - CPU/Wall Time
- Klein Tangent Features (task 7)



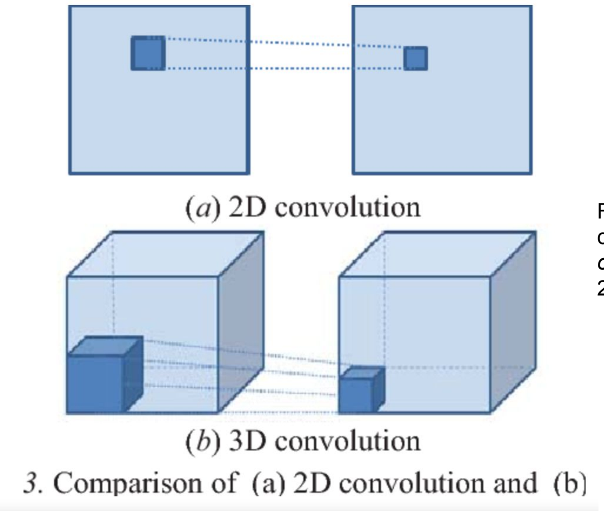
Latest Work

- Background
  - 3D Convolution
  - Klein Tangent Features
- Results
  - Core Metrics
  - Comparison of Models

3D Convolution - Background



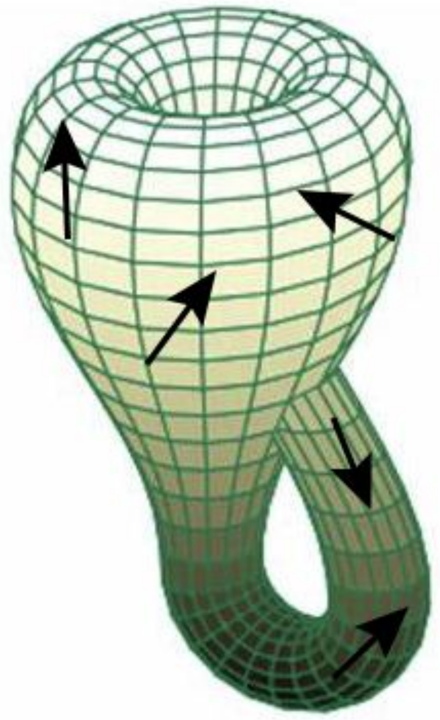
Tran, Du, et al. "Learning spatiotemporal features with 3d convolutional networks." *Proceedings of the IEEE international conference on computer vision*. 2015.



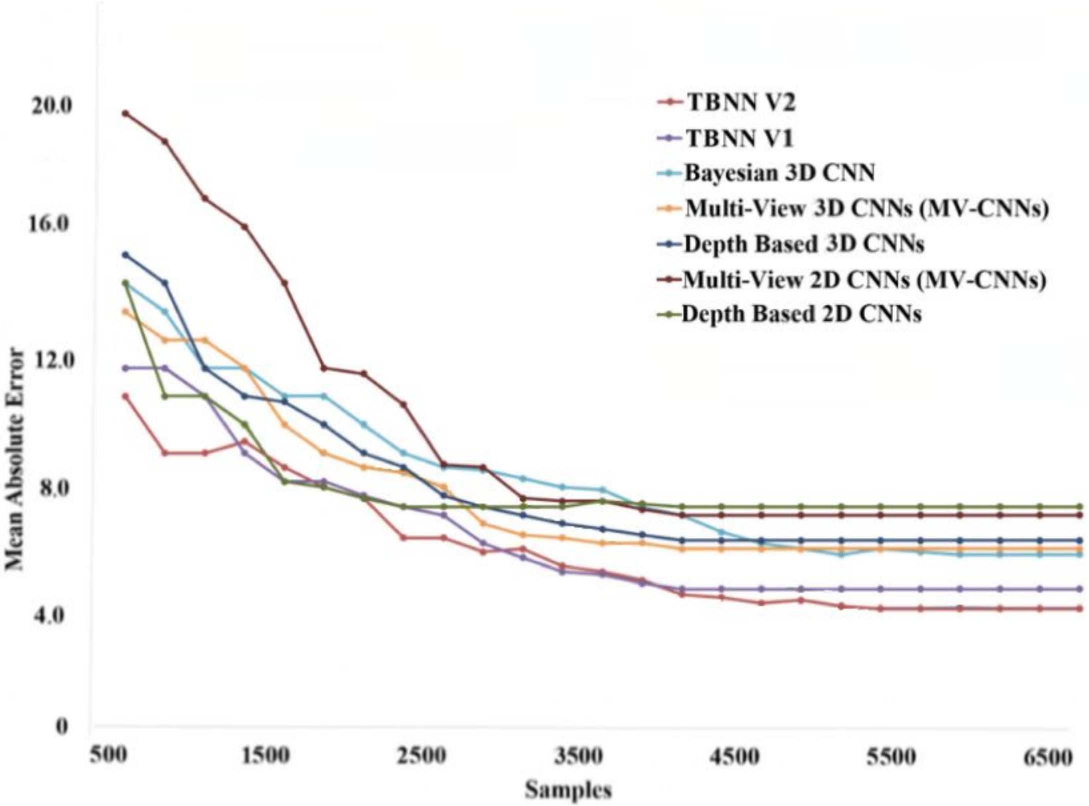
Fan, Lei, et al. "Lung nodule detection based on 3D convolutional neural networks." *2017 International conference on the frontiers and advances in data science (FADS)*. IEEE, 2017.

Klein Tangent Features

- Space of features parameterized by tangent bundle of the translational Klein Bottle.
- Translational Klein Bottle is a 3D manifold, and the tangent bundle is 6 dimensional.
- Intuition:
  - An image patch in the embedded Klein bottle has a natural 'orientation' given by the angle theta
  - One sees lines through the center of the image at angle theta + pi/2. There is a 2d image path along the line perpendicular to the lines in the image.
  - Video is same concept: Videos that change in time are obtained by enlarging the Klein bottle to its tangent bundle.
  - The tangent bundle consists of pairs of a point theta1, theta2, r and a vector (u,v.w) tangent to theta1, theta2, r (number of units along the line)
  - There is an embedding that sends such a pair to a video patch that a a fixed time is an image.



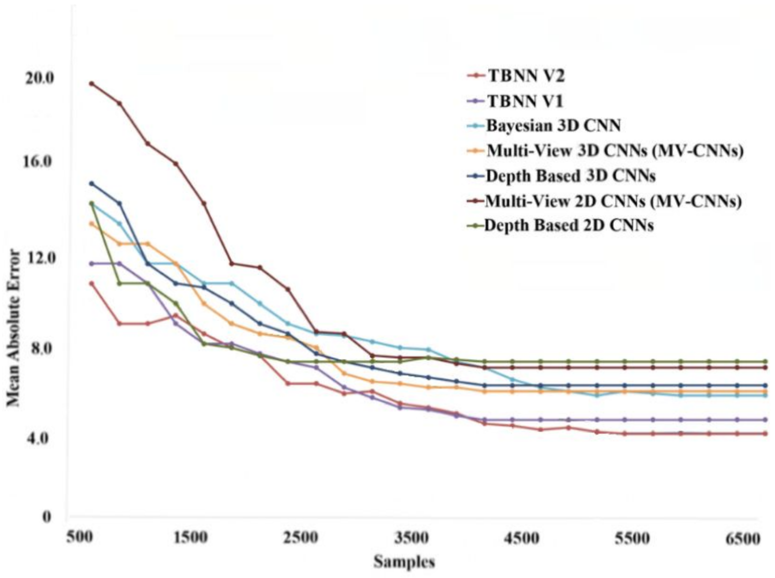
Model Performance



Task	Model	Mean Absolute Error (MAE)	Model Training Time (minutes)	Samples to Convergence (n=6500, round numbers), “asymptotic relative efficiency”	Samples for Top model to reach converged max performance of comparison model	Data Efficiency relative to top model (
Ejection Fraction (EF)	Human Experts	3.12	NA	NA	NA	NA
EF	TBNN V2	5.6	311	5750 samples to converge (at least 2 iterations of stability)	5750	5750/5750 = 100%
EF	TBNN V1	6.1	331	4750	4250	4250/4750 = 89%
EF	Bayesian 3D CNN	7.0	424	6250	3000	3000/6250 = 48%
EF	MV CNN 3D	7.2	387	4750	2500	2500/4750 = 53%
EF	DB CNN 3D	7.4	365	4750	2250	2250/4750 = 47%
EF	MV CNN 2D	8.3	321	5000	2000	2000/5000 = 40%
EF	DB CNN 2D	8.8	248	4500	2000	2000 / 4500 = 44%

Averaged Results (10 runs)

- TBNN V1, V2 outperform in core metrics
  - Sample Efficient
  - Most performant by mean absolute error
  - TBNN V1 is highest performer based on sample weighted MAE
- Worked
  - Performant models
  - Relatively fast training times
  - Efficient use of data
- Didn’t Work
  - Interpretation
  - Unclear optimal conditions for hybrid models
  - Unclear value of DL upgrades to TBNNs
- Main Takeaway
  - Method shows sufficient promise for further study



Future Work

- Increase number of options for Klein bottle features for video
  - Textures, Grid Sizes, bulls-eye and polka dot
- Attention
  - Study and implement attention in current design
  - For static images and video clips
- Visualization of results
  - Uncertainty bounds for predictions
  - Bayesian TNNs?
- Theory and Deployment: Understand conditions for success and failure for TNNs
  - What are good TNN problems?
  - How can TNNs and DNNs complement each other?
  - Different conditions include, e.g., kinds of data, quality of data, active, few shot, real time, federated learning, etc.

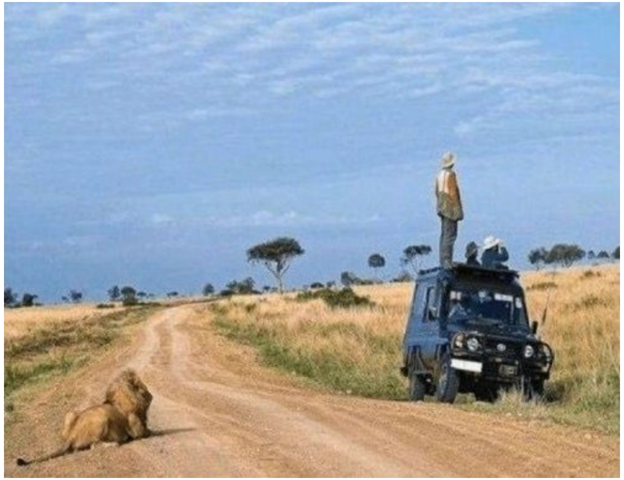
Appendix

Future

- Attention
    - Moving object detection and identification problem
    - In lieu of explicit outlines, use idea of attention
    - Labeled Data
        - Error Prone
        - Expensive and limited
        - Forces solving problem in particular ways

Attention

- Black and White images
  - Vector of length equal to number of pixels with 0/1 entries
  - Attention considers as set of pixel position pairs
  - In binary case, can reconstruct image from collection of pixels with value 1
  - Represents a compression and indicates set can be used as input / output
- Color
  - Threshold from below to create sets
  - Use any feature to threshold
- Video
  - Local motion, such as present in the video feature collection
  - Potentially powerful tool

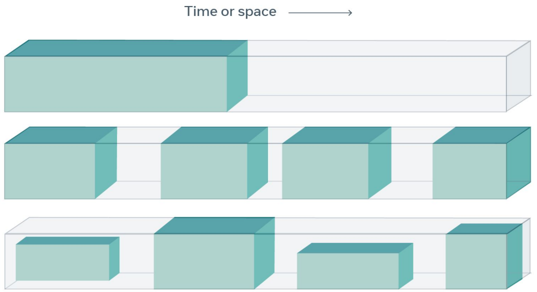


Self Supervision

- Supervisory signals from the data, using the underlying structure in the data.
- Goal is to learn representations invariant under different distortions
  - E.g. Data Augmentation
  - Different, noisy datasets
- Exploring phenomenology and extracting information captured in correlations, metadata embedded in the data, or domain knowledge
  - E.g. ejection fraction is not most important thing for cardiologists
  - Can create common categories of outcomes (predisposition, asymptomatic, admission to hospital of death)

Self Supervised Learning (SSL) (cont.)

- Core SSL Task is to predict missing parts of the input
- “The (blank) chases the (blank) in the savanna”
  - System must learn lions and cheetahs can chase antelope or wildebeest but cats chase mice in the kitchen, not the savanna.
- Unclear how to represent uncertainty in the prediction for images
  - When missing words can associate a score - e.g. high score for lion, cheetah and low score for other words
  - Missing patches in an image or frames in video - e.g. list all the possible frames and score the options?



https://scontent-sjc3-1.xx.fbcdn.net/v/t39.2365-6/148954125_461761118405979_2035914075893596810_n.png?_nc_cat=107&ccb=1-5&_nc_sid=ad8a9d&_nc_ohc=rp3TDnEDa4oAX-cUEA9&_nc_ht=scontent-sjc3-1.xx&oh=00_AT-ao1yAPazK7aWi9DvGQ_1iqcK0dKElieXppH5oigTg0Q&oe=6236E859