Session iv: Advanced topics

Greg Tsagkatakis

CSD - UOC

ICS - FORTH

Overview

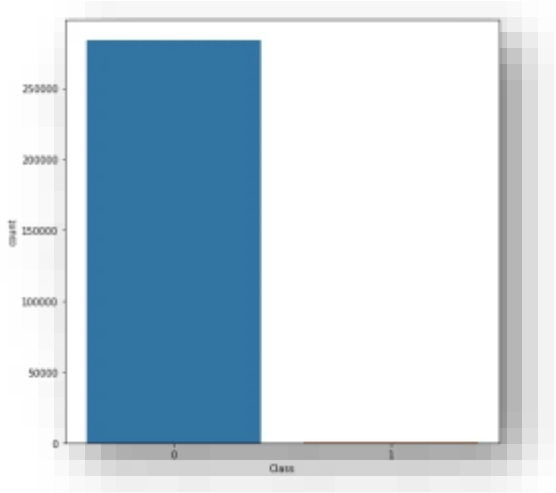
Class imbalance

Deep Reinforcement Learning

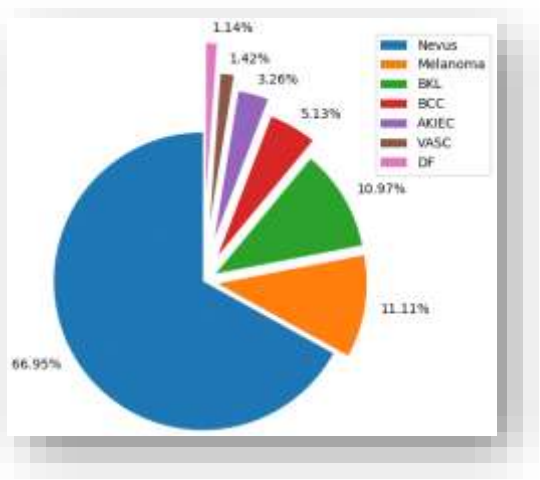
Hardware-in-the-loop

Beyond SotA

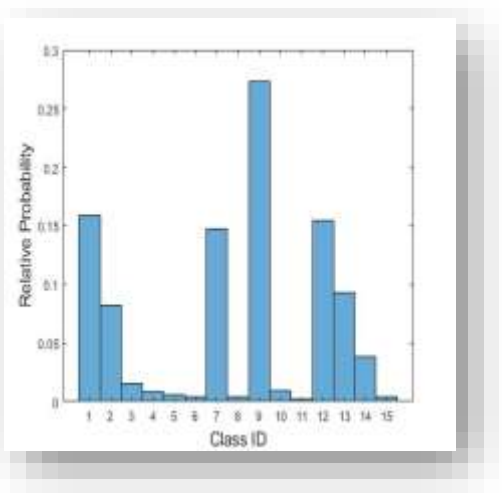
Challenge of imbalanced datasets



Fraudulent transactions



multi-source dermatoscopic images of 7 skin lesion



Land Cover \ Land Use

Problem Statement

Class Imbalance: Why is this a problem?

A disproportionate ratio of observations in each class for a given dataset

What causes imbalanced classification?

i. Properties of the domain (Natural distribution of classes, difficulty of collecting samples from certain classes)

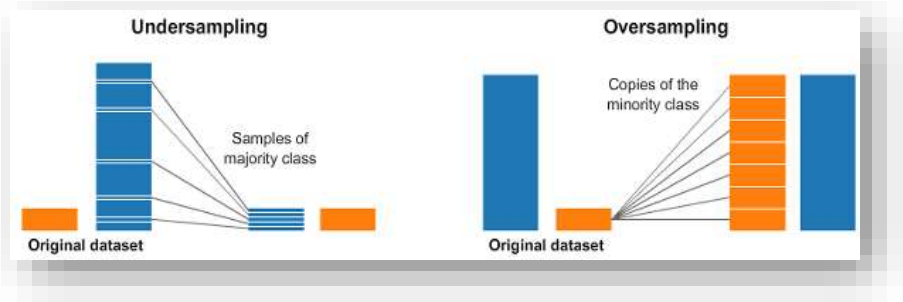
ii. Data sampling

1. Most algorithms assume equally distributed data & tend to be more biased towards the majority class -> Unstable predictions & bad classification of the minority class.

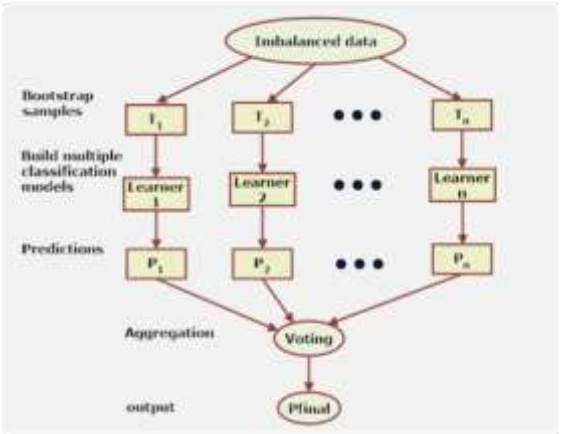
2. Affects many real world-problems (biomedical imaging, spam, anomaly detection, etc.)

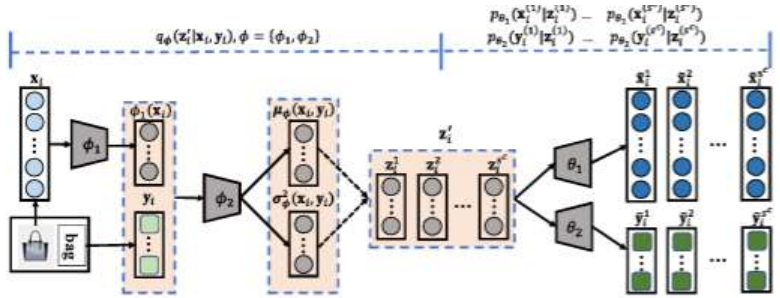
3. Challenge to be addressed

State-of-the-art



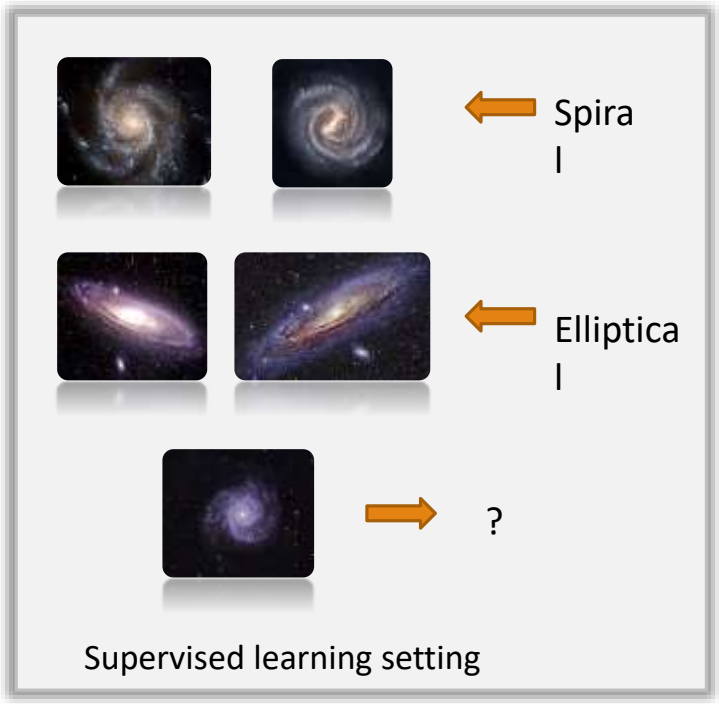
Nitesh Chawla, et al. "SMOTE: Synthetic Minority Over-sampling Technique." Journal of Artificial Intelligence Research (2002)

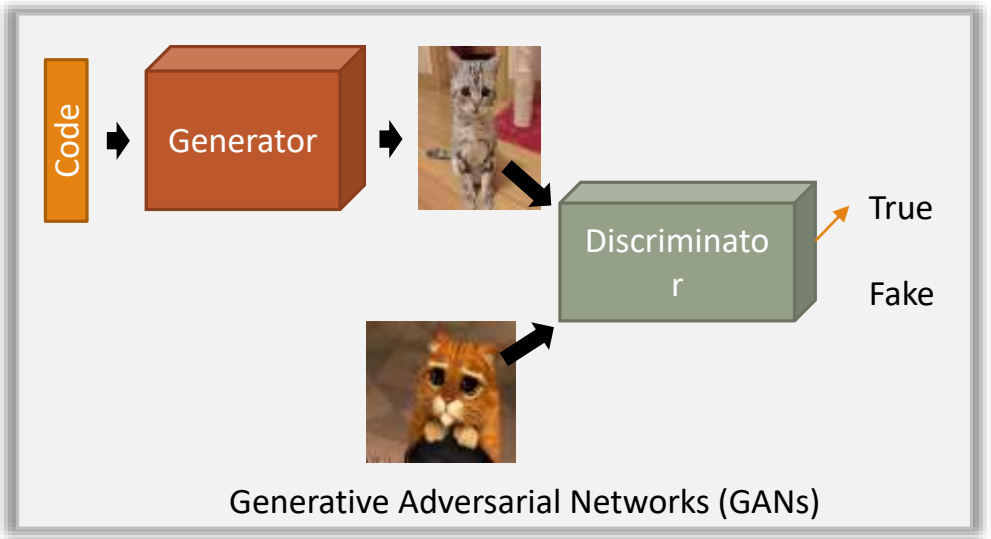


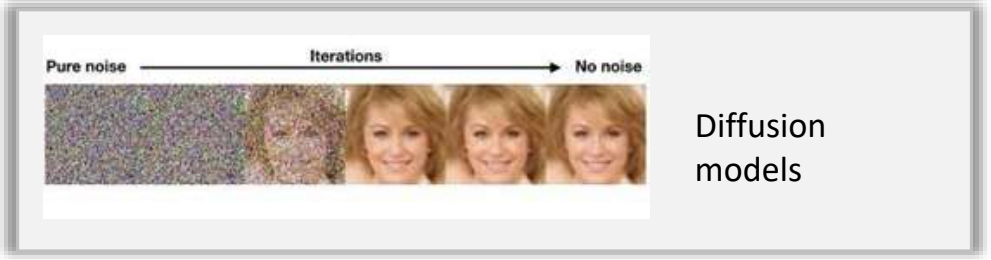


Xinyue Wang, et al. "Deep Generative Model for Robust Imbalance Classification." IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)

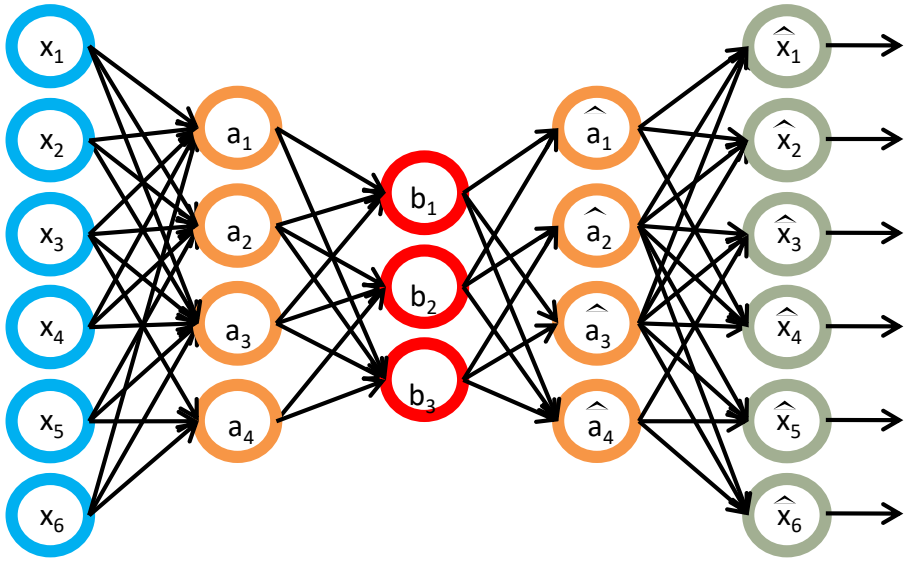
Generative models

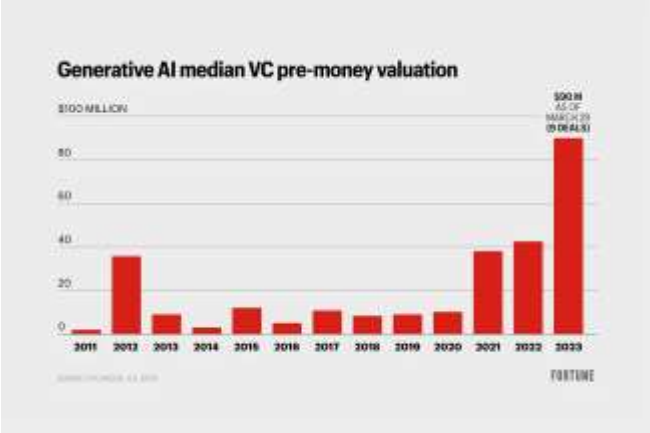






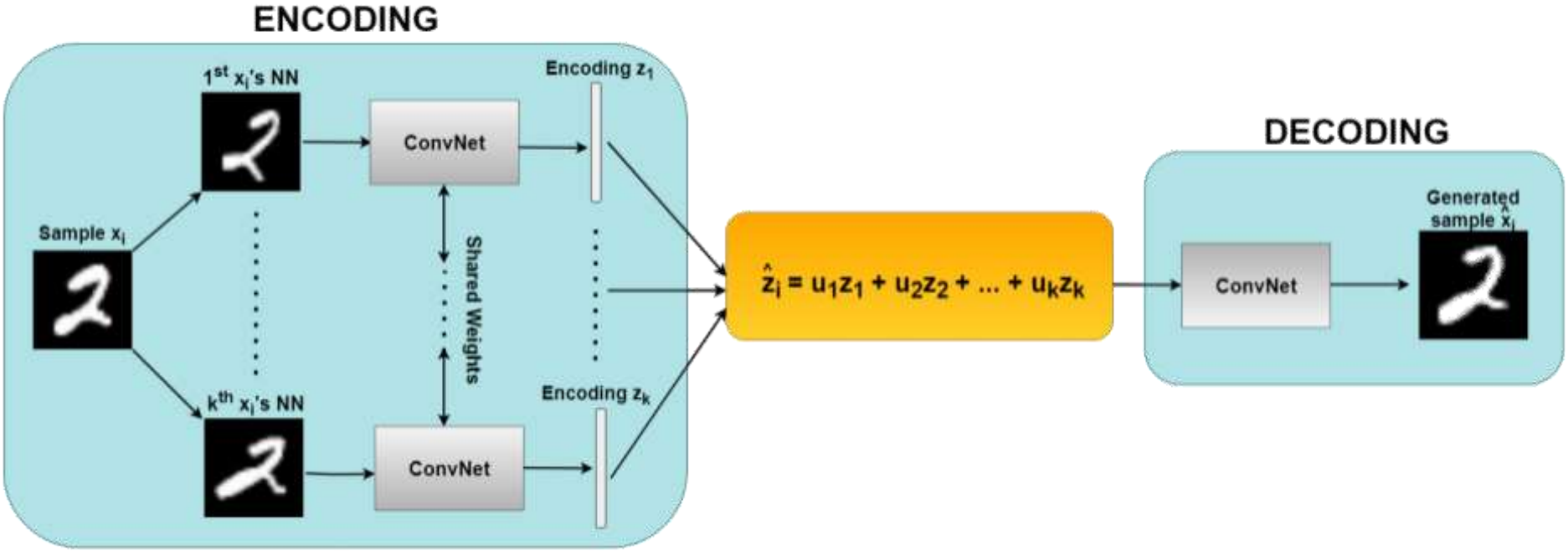
Autoencoders





$$loss = \frac{1}{m}\sum_{i=1}^{m}\|\hat{x}_i - x_i\|_2 = \frac{1}{m}\sum_{i=1}^{m}\|\hat{a}_1(b_1(a_1(\hat{x}_i))) - x_i\|_2$$

GENDA



$$L = \frac{1}{M}\sum_{i=1}^{M}(x_i - \hat{x}_i)^2 = \frac{1}{M}\sum_{i=1}^{M}(x_i - d(e(N(x_i))))^2$$

Troullinou, E., Tsagkatakis, G., Losonczy, A., Poirazi, P., & Tsakalides, P. A Generative Neighborhood-based Deep Autoencoder for Robust Imbalanced Classification. *IEEE Transactions on Artificial Intelligence,* 2023.

Formulation

Encoder Model (e) **:** $p(\hat{z}_i|N(x_i)) \equiv p(\hat{z}_i|x_1, x_2, \dots, x_k) \quad \forall\, i = 1, \dots, M$

where M: number of samples, k: number of Nearest Neighbors ($k = 2$)

$$\hat{z}_i = \sum_{j=1}^{k} u_j \cdot z_j \qquad \forall\, i = 1, \dots, M\,, \quad u_j \in \mathcal{U}(0,1) \quad \& \quad \sum_{j=1}^{k} u_j = 1$$

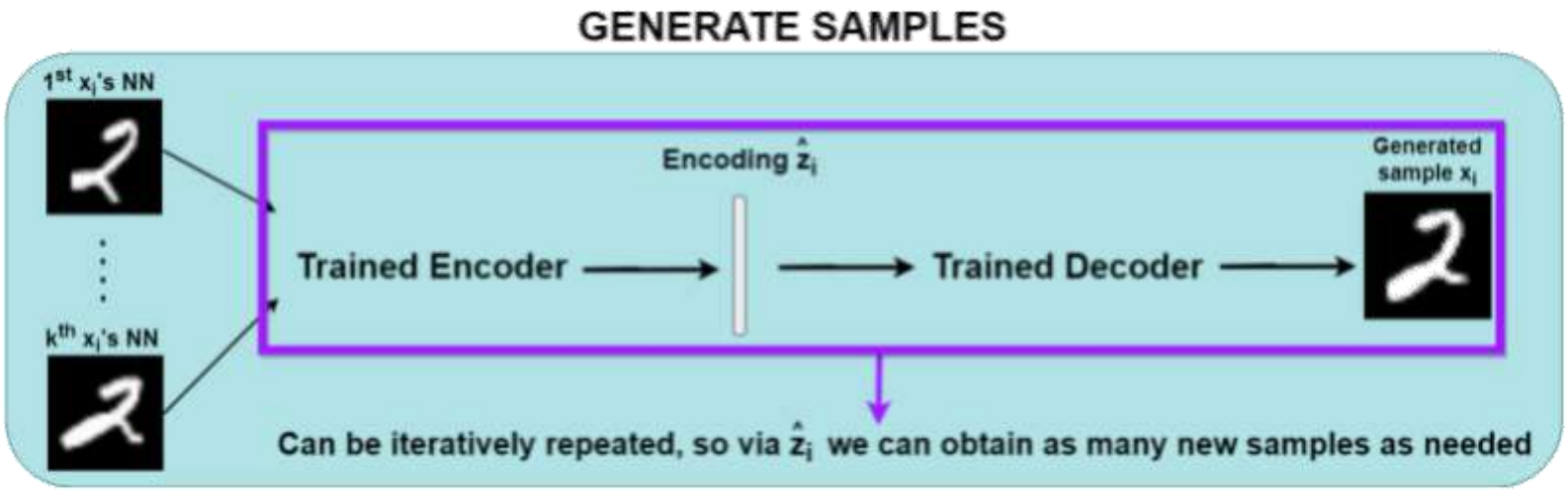
$$z_j = f\left(W h_j + b\right) \qquad \forall\, j = 1, \dots, k$$

Decoder Model (d) **:** $q(\hat{x}_i|\hat{z}_i) \quad \forall\, i = 1, \dots, M$

$$\hat{x}_i = \sigma(W' h + b') \qquad \forall\, i = 1, \dots, M$$

$$\mathcal{L} = \frac{1}{M}\sum_{i=1}^{M}(x_i - \hat{x}_i)^2 = \frac{1}{M}\sum_{i=1}^{M}\left(x_i - d\left(e\left(N(x_i)\right)\right)\right)^2$$

Results



	HAR			TwoLeadECG			Ca^{2+} Imaging		
Method	ACSA	F1-Score	Precision	ACSA	F1-Score	Precision	ACSA	F1-Score	Precision
Baseline	0.605	0.536	0.555	0.5	0.342	0.26	0.65	0.674	0.714
SMOTE	0.731	0.682	0.652	0.81	0.823	0.815	0.77	0.78	0.792
TimeGAN	0.713	0.67	0.643	0.735	0.716	0.693	0.697	0.674	0.654
GENDA	**0.877**	**0.878**	**0.883**	**0.829**	**0.838**	**0.817**	**0.787**	**0.797**	**0.809**

	HAR		
Rebalancing Approach	ACSA	F1-Score	Precision
Oversampling	0.642	0.645	0.65
Undersampling	0.53	0.52	0.525

AI in space

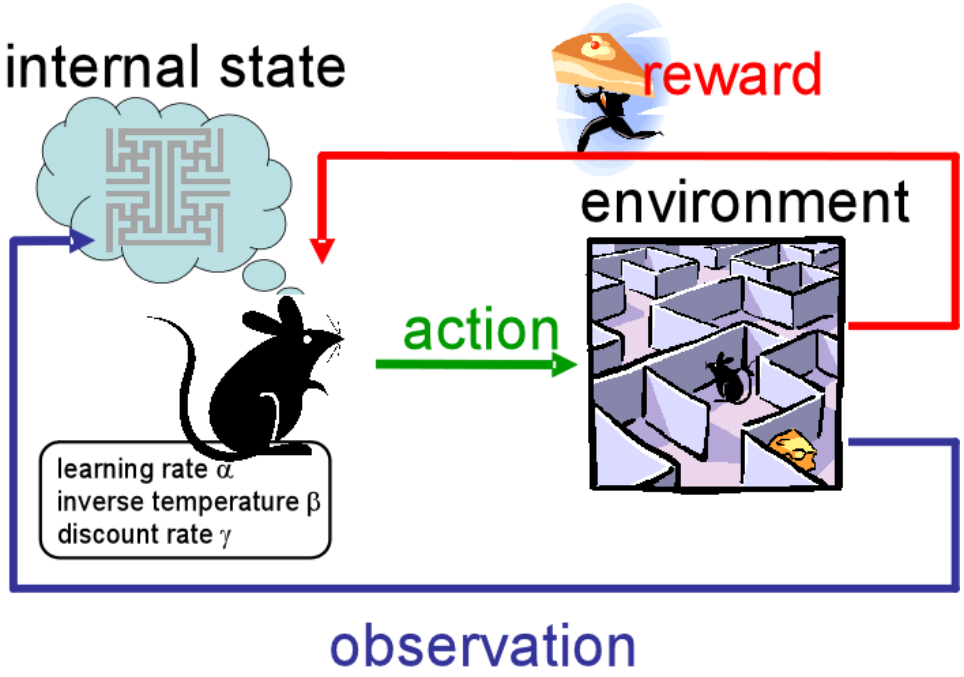
1st degree is when AI is optimizing pipelines that are traditionally executed on the ground like selecting which image is cloud-free and thus should be downloaded to the ground.

2nd degree is when AI algorithms are executed onboard in order to extract information that is much smaller in size and can thus be transferred to the ground by low-capacity telemetry channels.

3rd degree is when AI acts as an enabler for designing new missions which offer never-seen-before capabilities. In this case, on-board AI is introduced for maximizing the amount of information that can be captured through autonomous decision-making.

Reinforcement learning

Reinforcement learning: system interacts with environment and must perform a certain goal without explicitly telling it whether it has come close to its goal or not.



Types of Reinforcement Learning

Search-based: evolution directly on a policy

- E.g. genetic algorithms

Model-based: build a model of the environment

- Then you can use dynamic programming
- Memory-intensive learning method

Model-free: learn a policy without any model

- Temporal difference methods (TD)
- Requires limited episodic memory (though more helps)

Q-learning

- The TD version of Value Iteration
- This is the most widely used RL algorithm

Q-Learning

Define **Q*(s,a):** "Total reward if an agent in state **s** takes action **a**, then acts optimally at all subsequent time steps"

Optimal policy: $\boldsymbol{\pi^*(s)=\text{argmax}_a Q^*(s,a)}$

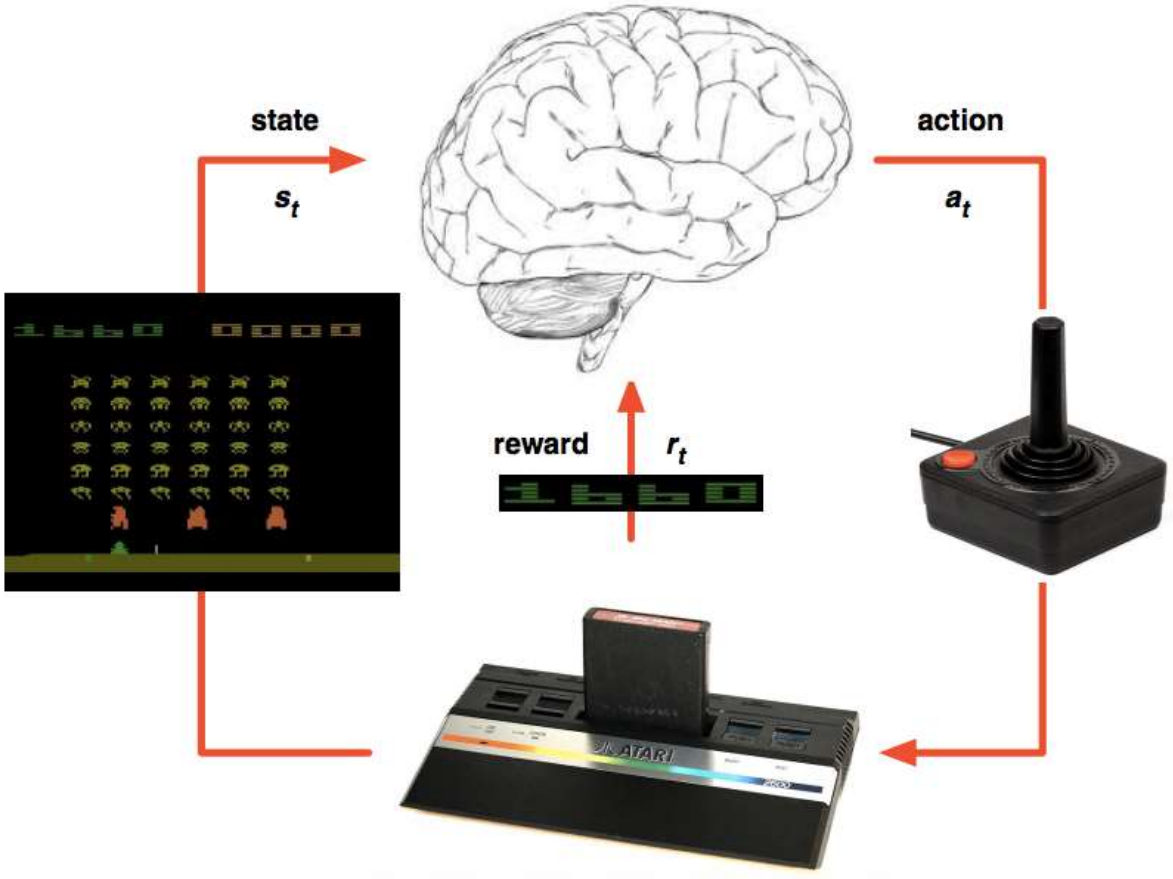
Q(s,a) is an estimate of **Q*(s,a)**

Q-learning motion policy: $\boldsymbol{\pi(s)=\text{argmax}_a Q(s,a)}$

Update Q recursively:

$$Q(s,a) = r + \gamma \max_{a'} Q(s',a') \qquad 0 < \gamma < 1$$

Deep Q-Learning



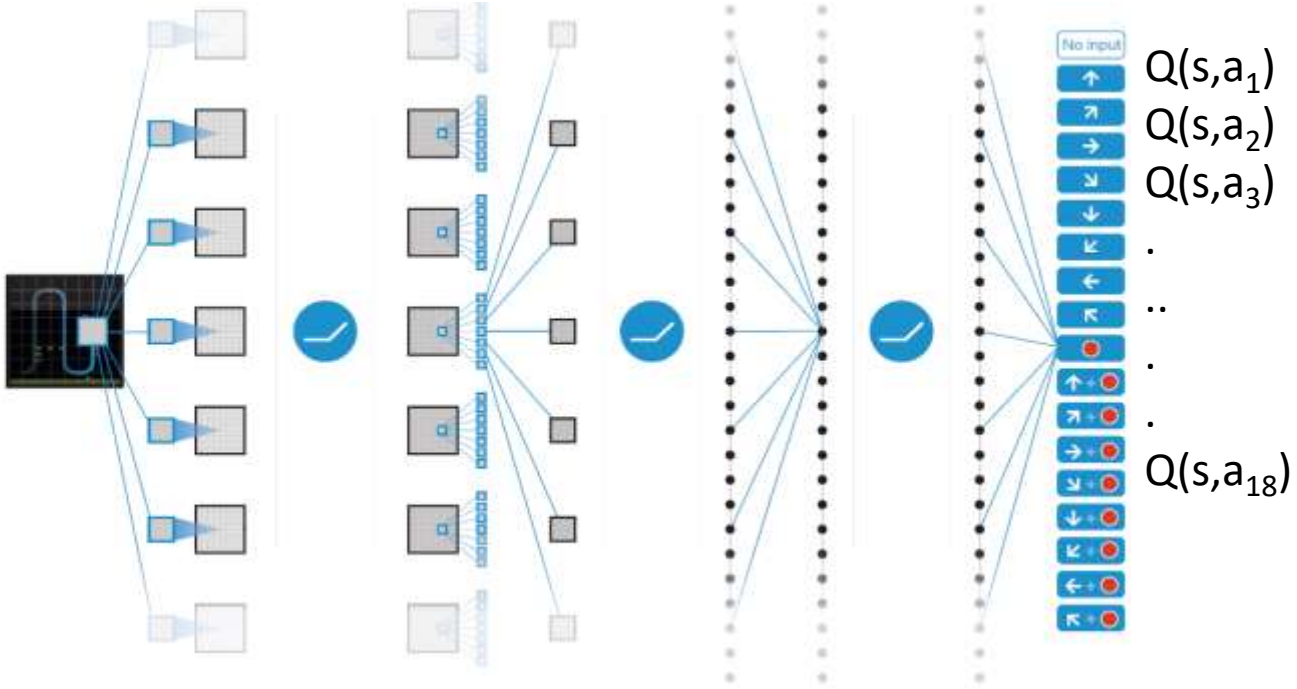
Mnih et al. Human-level control through deep reinforcement learning, *Nature* 2015

Deep Q learning in Atari

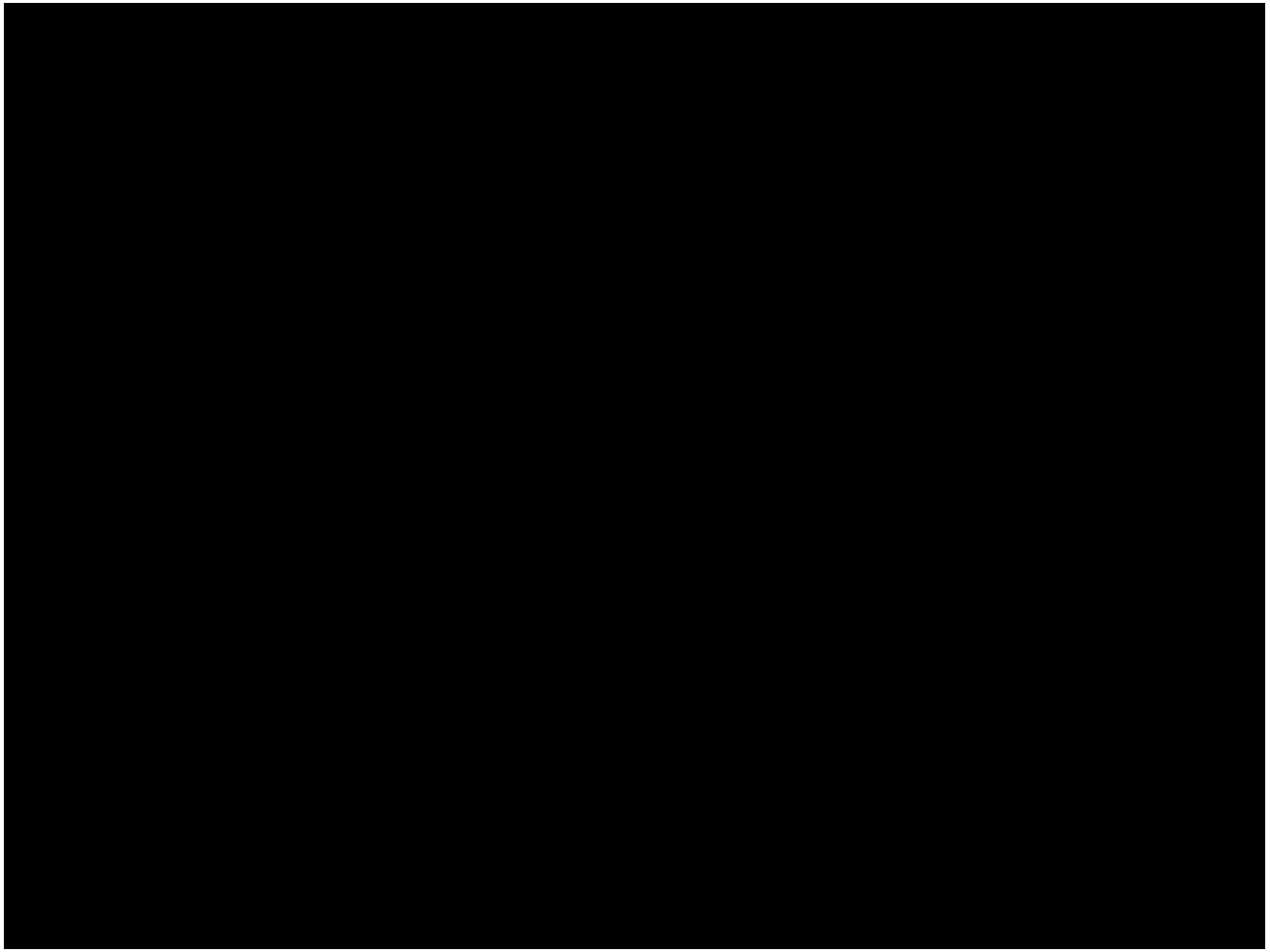
End-to-end learning of Q(s,a) from pixels s

Output is Q(s,a) for 18 joystick/button configurations

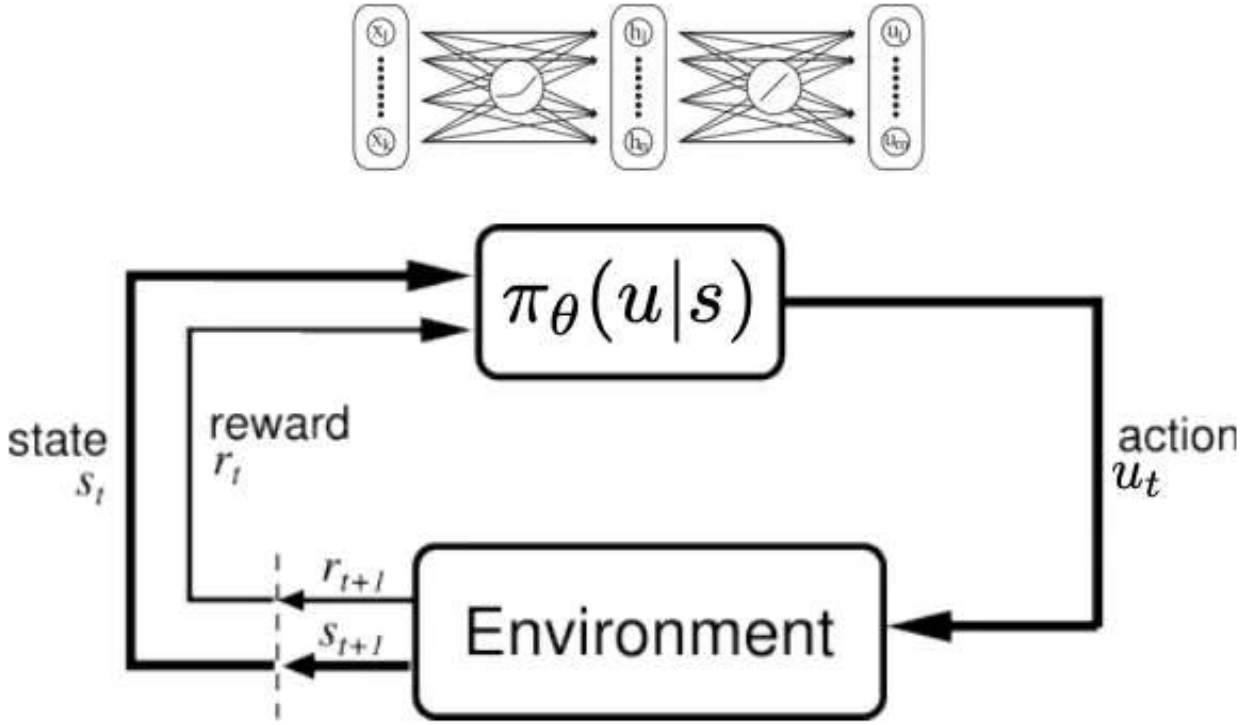
Reward is change in score for that step



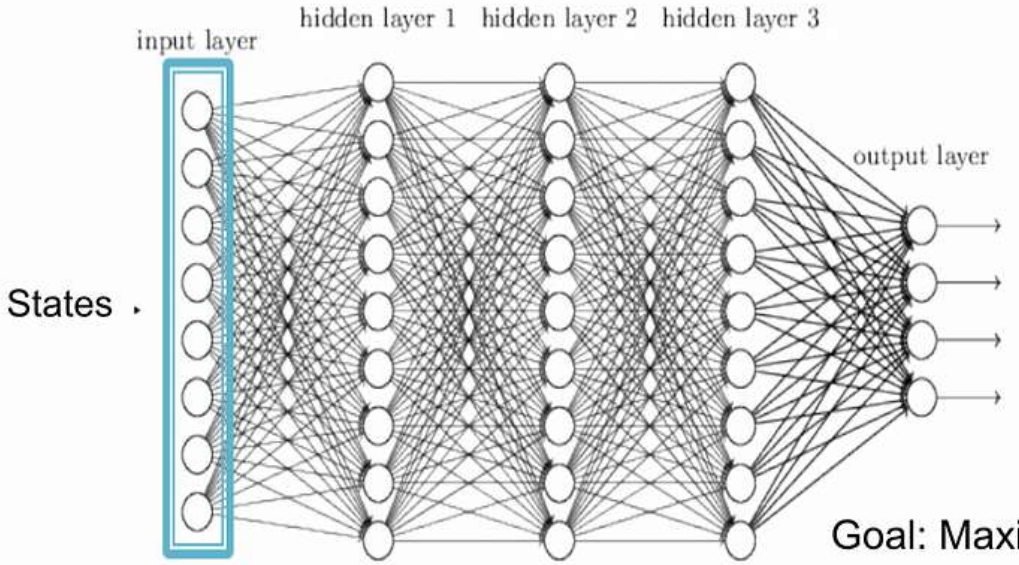
Breakout demo



Policy gradient approach



Deep Policy gradient approach



Goal: Maximize Expected Return J(θ)

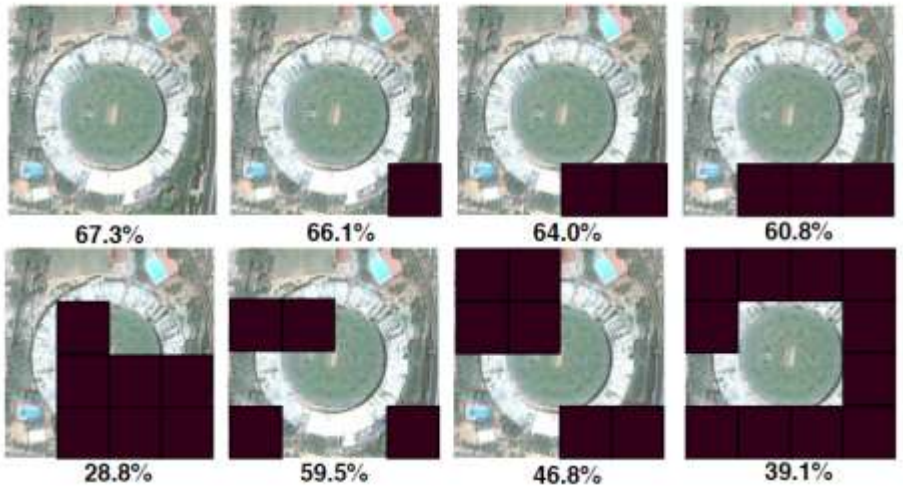
π_θ

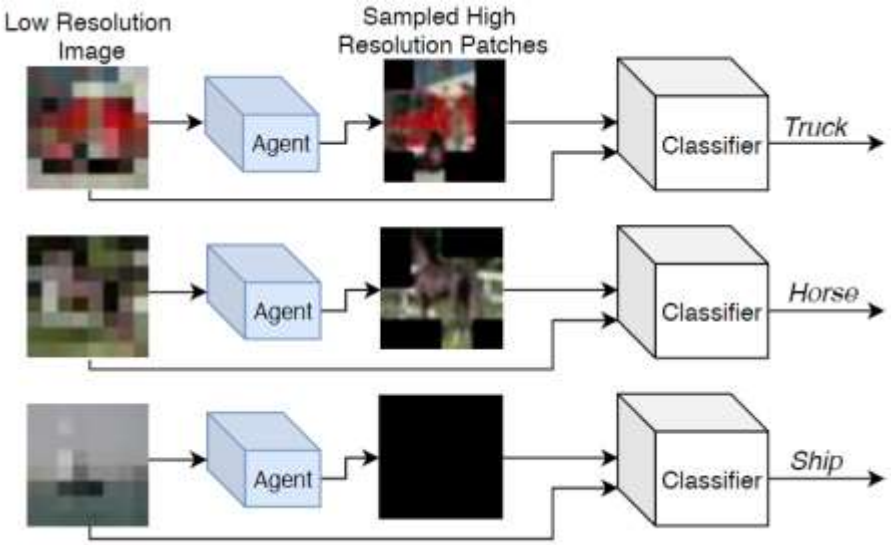
Policy Gradient

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \widehat{\nabla J(\boldsymbol{\theta}_t)}$$

Update policy parameter, θ using gradient ascent

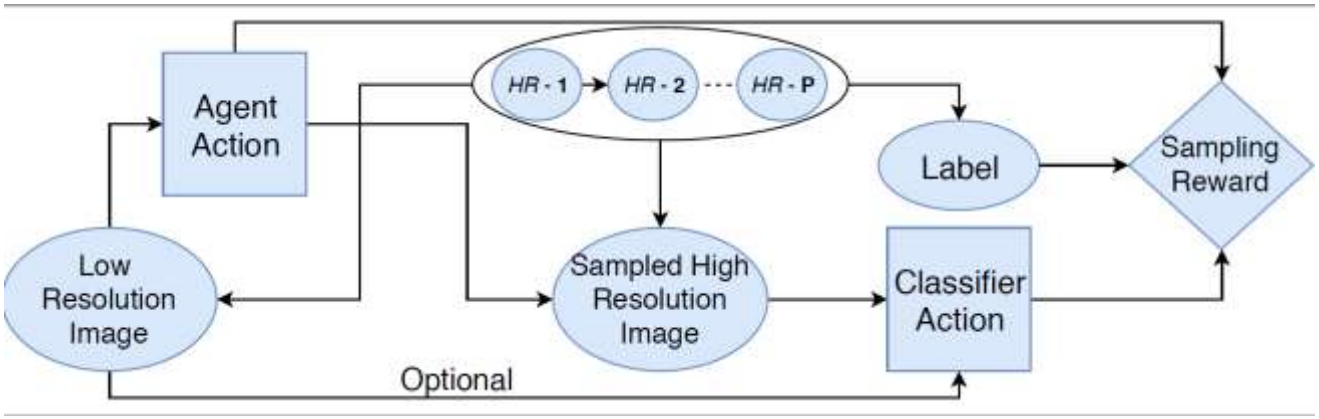
PatchDrop

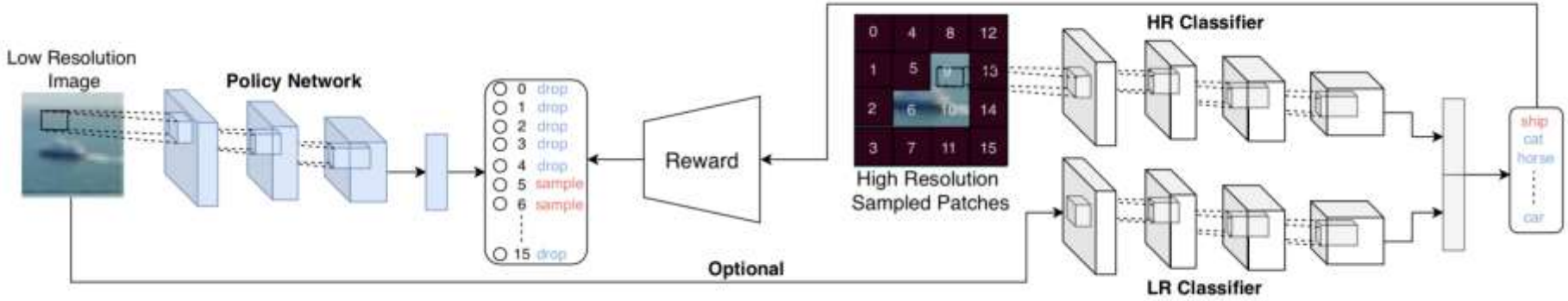




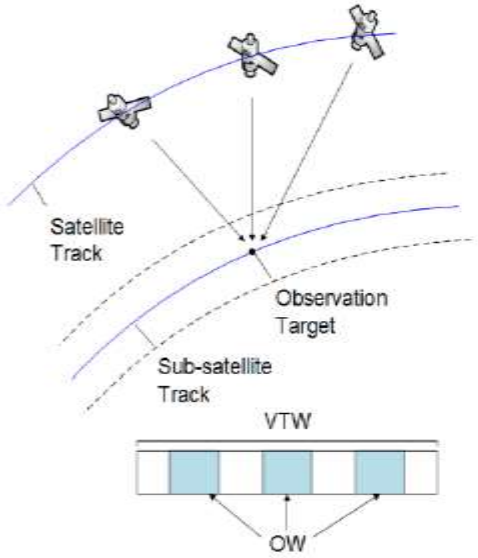
Uzkent, Burak, and Stefano Ermon. "Learning when and where to zoom with deep reinforcement learning." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020.

Deep reinforcement learning

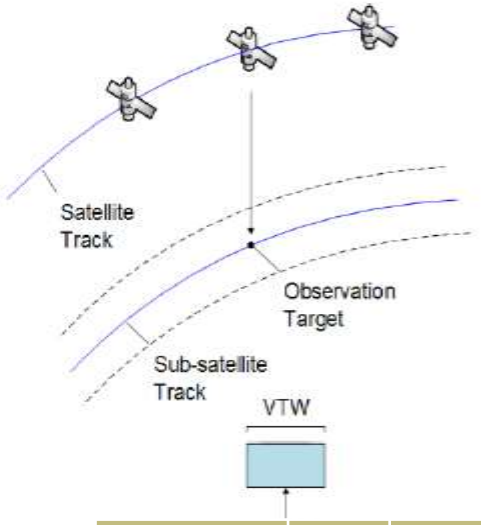




Agile Earth Observation Systems (AEOS)



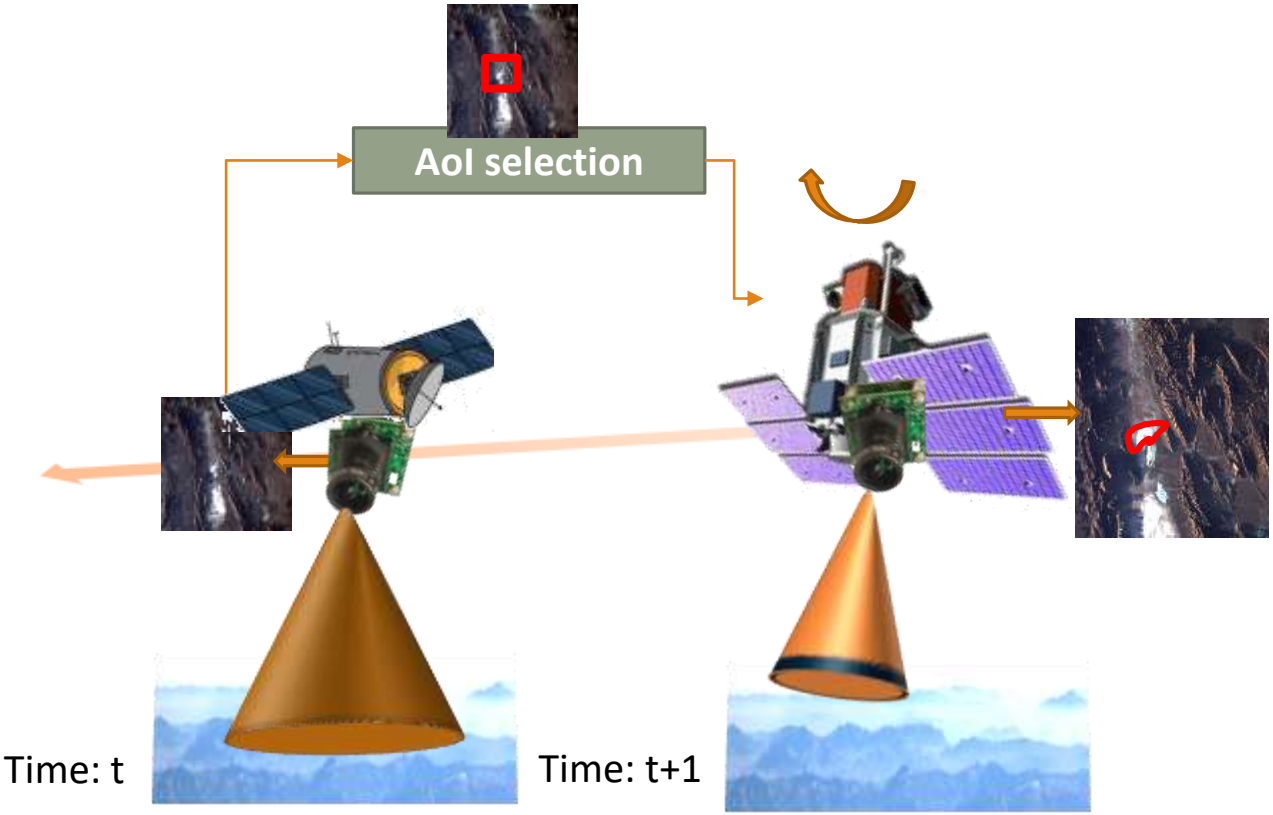
(a) agile satellite



Mission	Year	Resolution (m)	Swath (Km)	Revisit (days)
Sentinel 2 A/B	2015/ 7	10/20 MSI	290	10
Landsat 9	2021	15 (PAN), 30 (MSI)	185	16

Mission	Year	Country	Const.	Res. (m)	Swath (Km)
GeoEye-1	2008	USA	N	0.46-1.84	15
Pleiades	2011	France	Y	1-3	20
WorldView 1/2	2007 /9	USA	Y	0.3-3.7	17/14
SuperView -2	2022	China		0.3 (PAN) – 1.2 (MSI)	15

Dual Cam Project



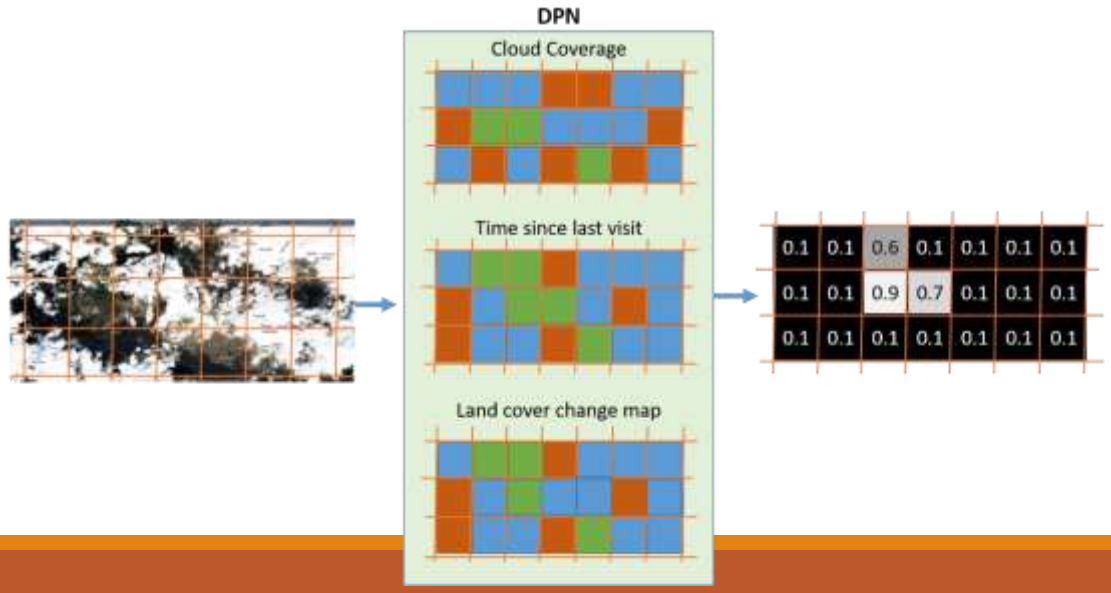
AI-on-board

Challenge: real-time performance constraints

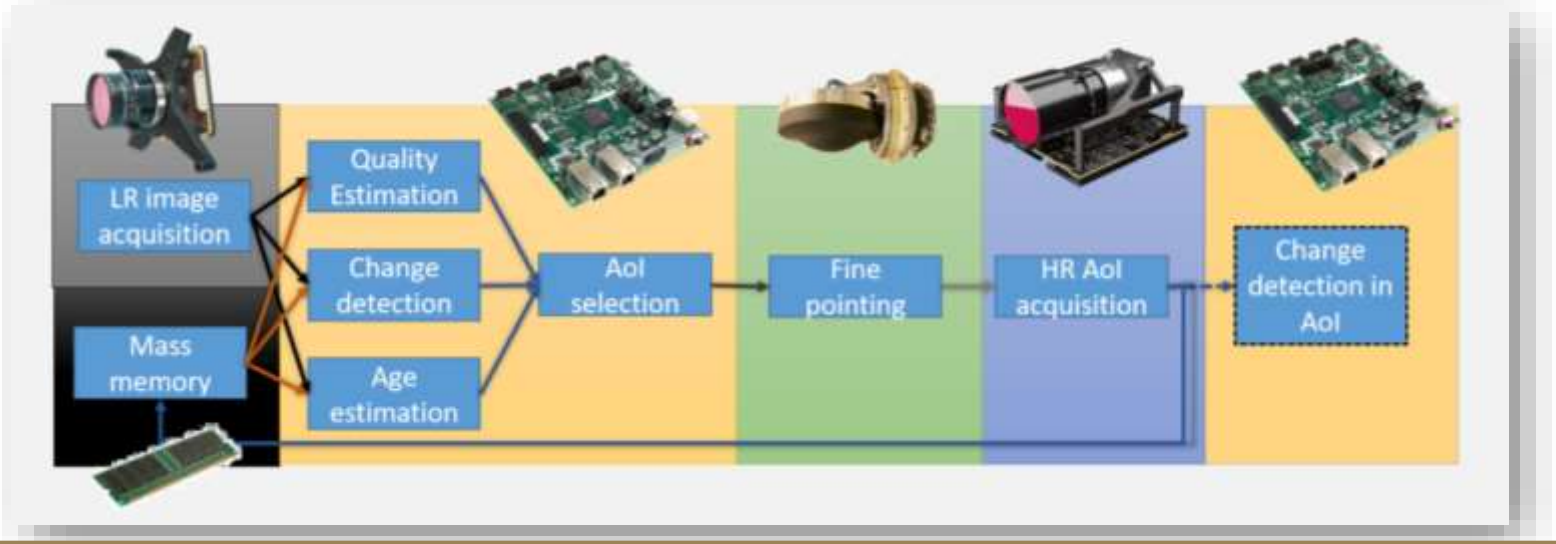
Key novelty: This concept is only made feasible if onboard AI is considered

Deep Policy Network (DPN)

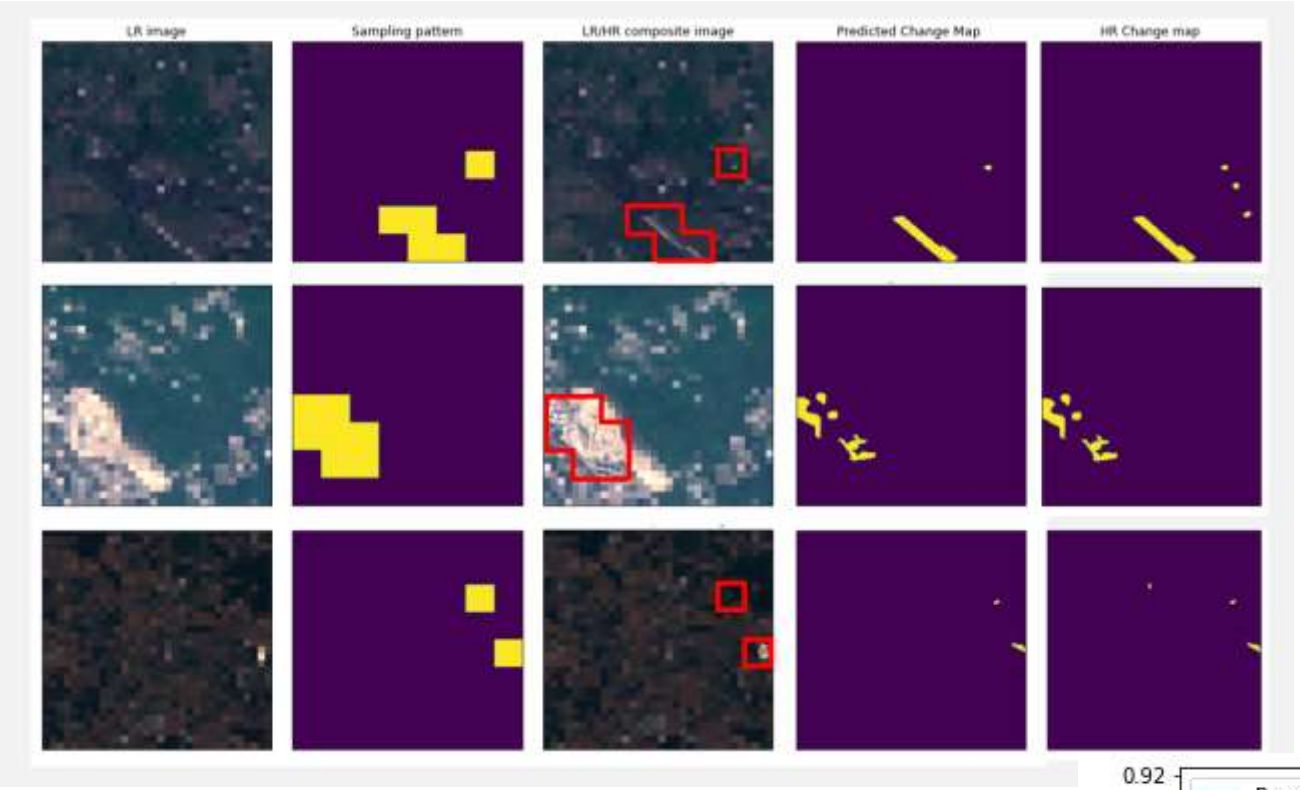
- Analyze the LR observation from the first camera
- Associate with each region an estimated value
- Specify action (AoI selection) to be taken

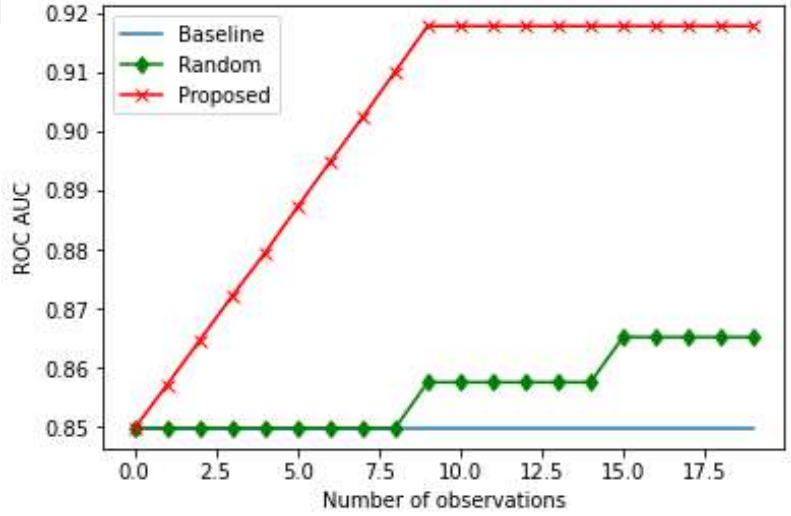


Realization



Initial LC class	New LC class	Domain
Sparsely vegetated areas	Construction sites	Urbanization
Forest	Burned area	Forest fires
Non-irrigated land	Permanently irrigated land	Agriculture
Green urban areas	Urban fabric	Sustainable Development
Semi-natural land	Wetland	Flooding
Natural and semi-natural land	Cropland	Biodiversity


LR image
Sampling pattern
LR/HR composite image
Predicted Change Map
HR Change map


Baseline
Random
Proposed
ROC AUC
0.92
0.91
0.90
0.89
0.88
0.87
0.86
0.85
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Number of observations

Object/target recognition

- Floods

- Oil spills

- Solar PV parks

- **Active fire**

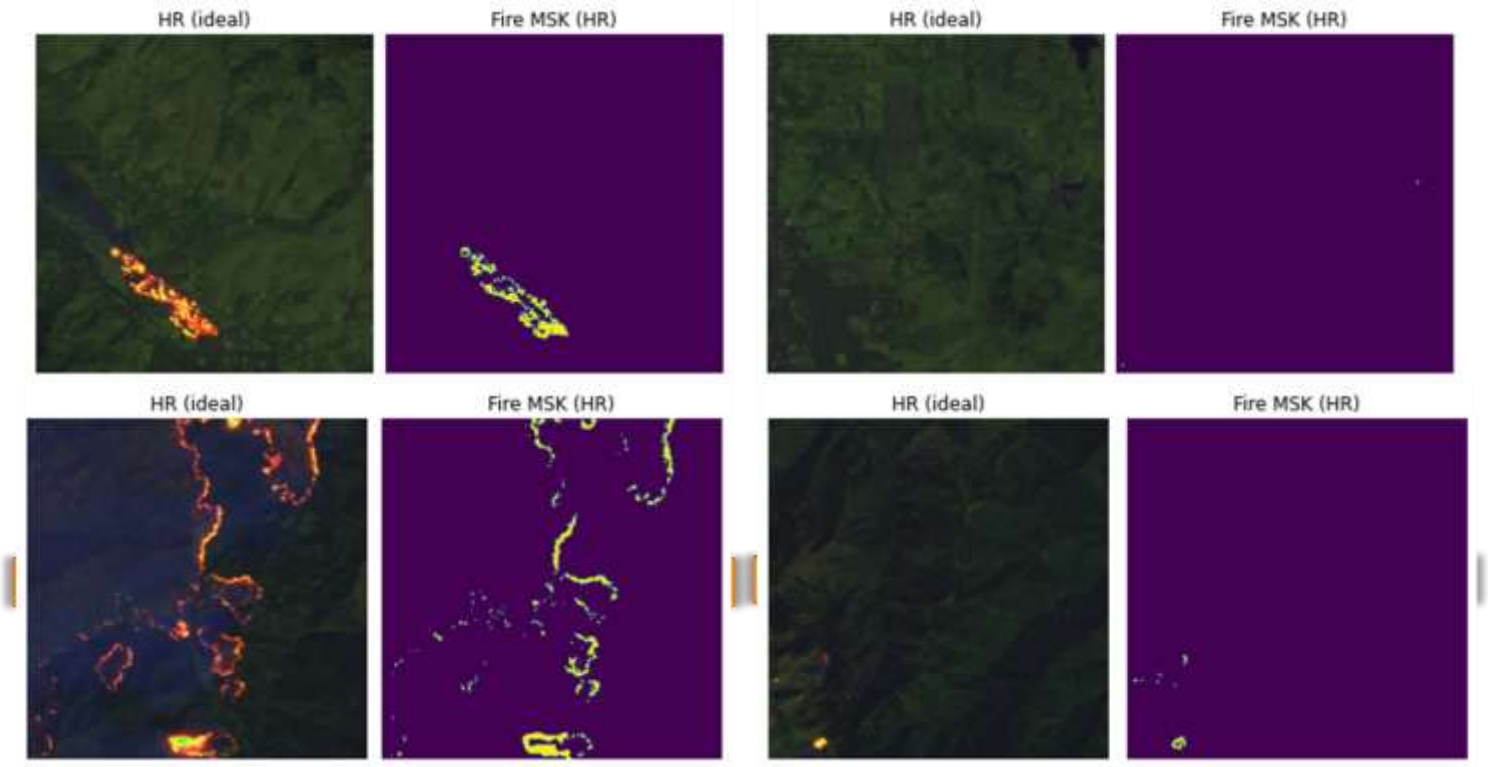
- Floating plastic litter (distribution)

- Volcanic eruption

Active fire detection dataset

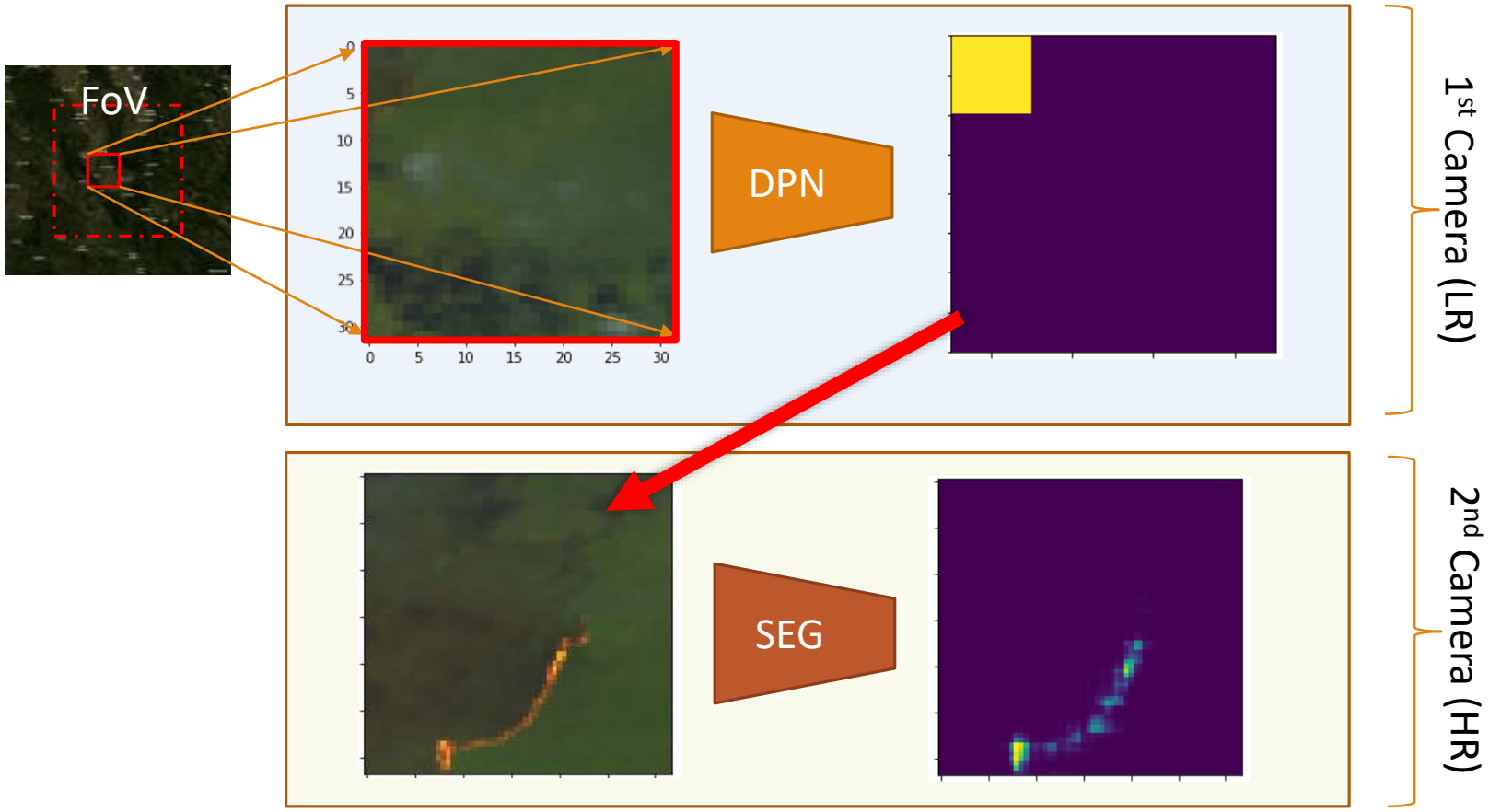
Input: Landsat 8, 3 band: 6(1570nm), 5(850nm), 1(430nm)

Labels: Manual annotations (pixel-level classification)



de Almeida Pereira, Gabriel Henrique, et al. "Active fire detection in Landsat-8 imagery: A large-scale dataset and a deep-learning study." *ISPRS Journal of Photogrammetry and Remote Sensing* 178 (2021): 171-186.s

System overview



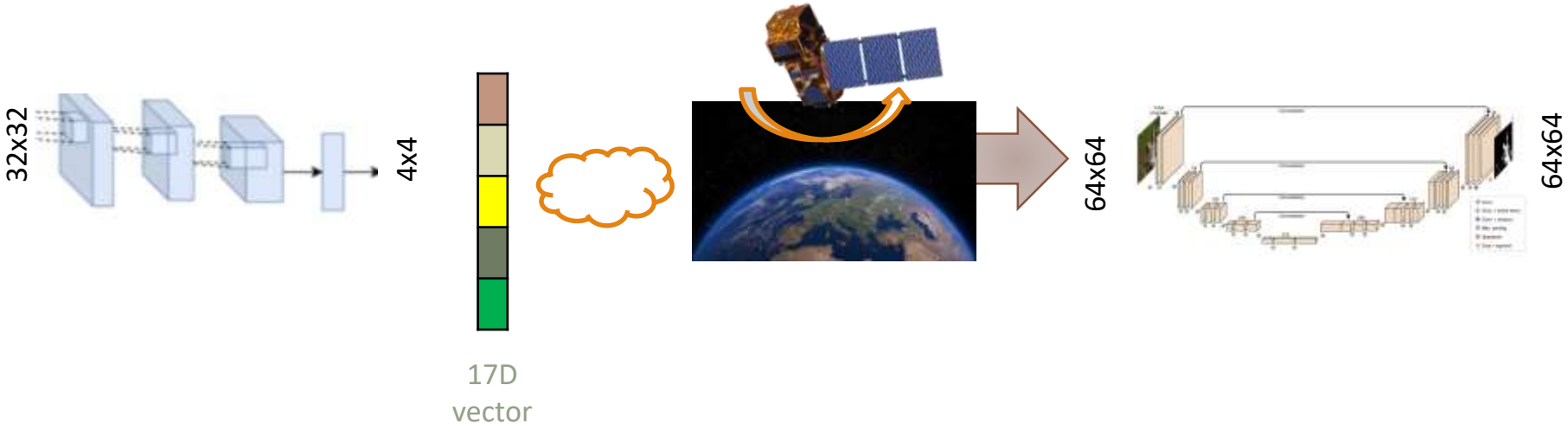
DNN

Deep Policy Network

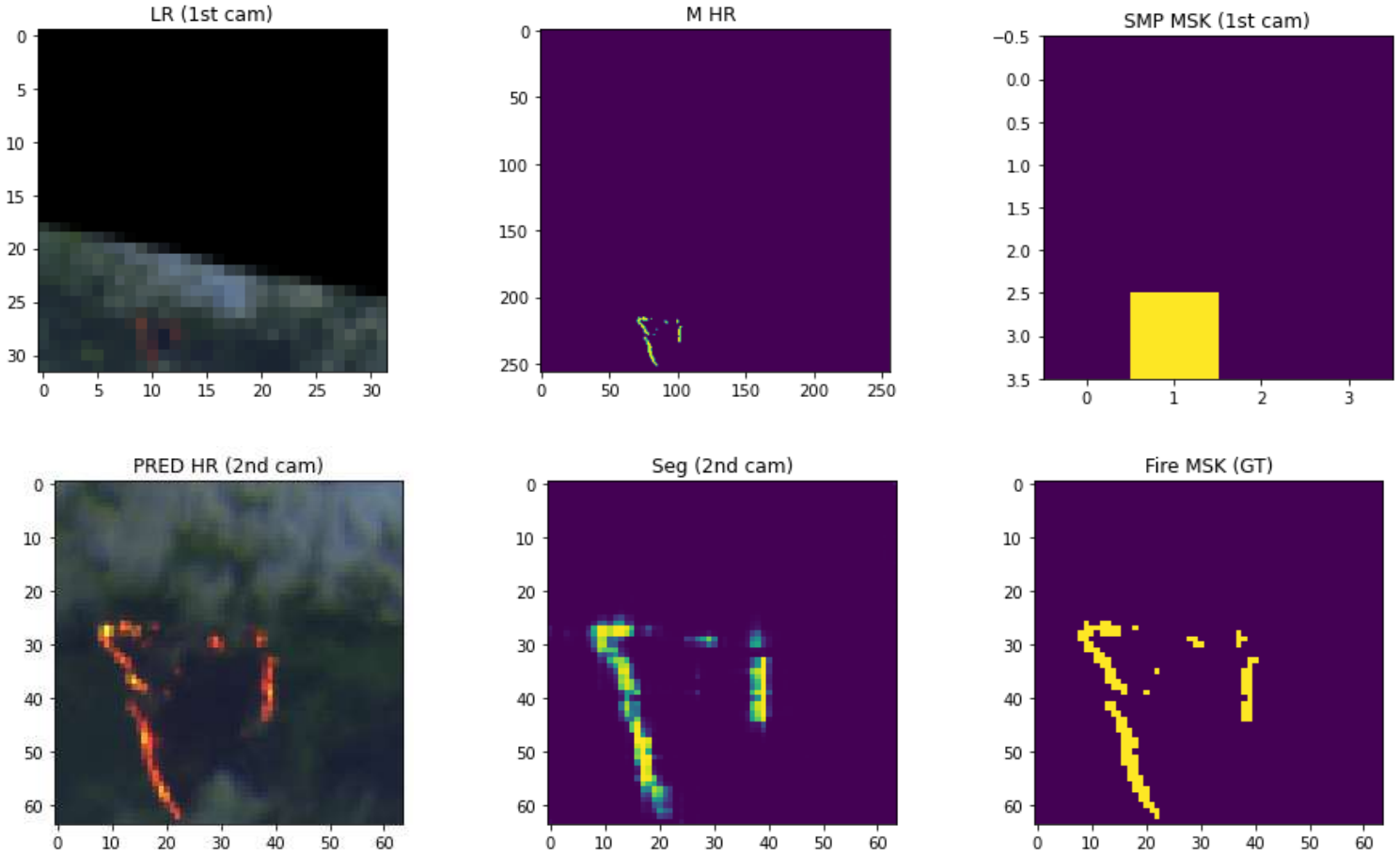
- Input: Low Res image
- Output: divide into 8x8=16 patches -> Select 0 or 1 patch per image

Active Fire Detection Network

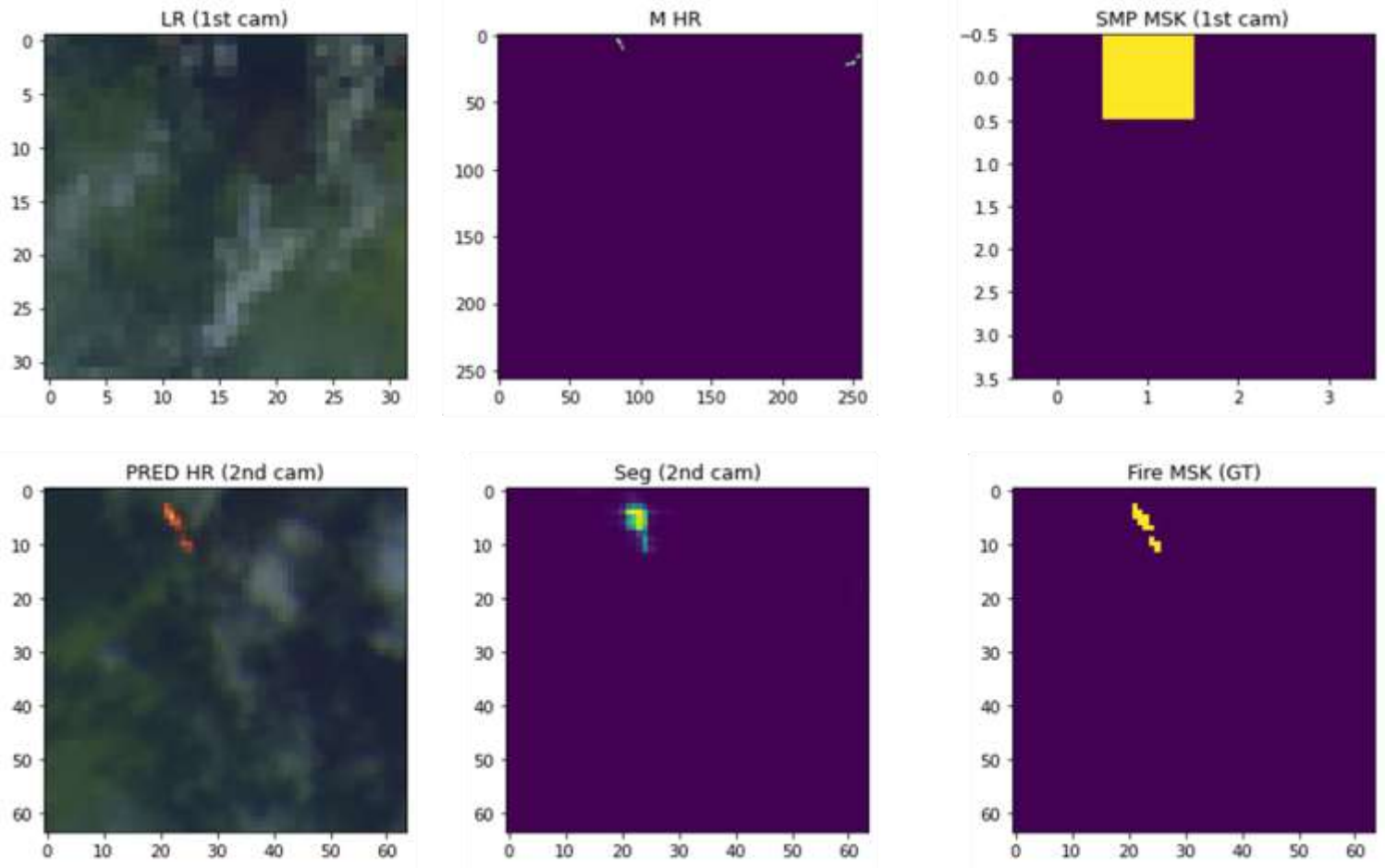
- Input: High Res patch
- Output: Active fire mask (per pixel classification 0/1)



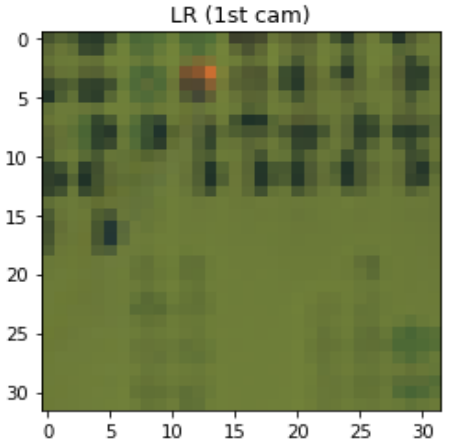
Performance

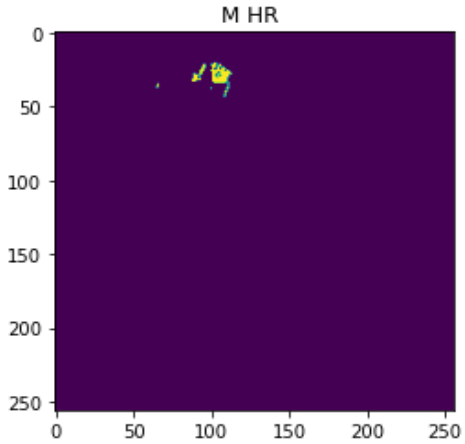


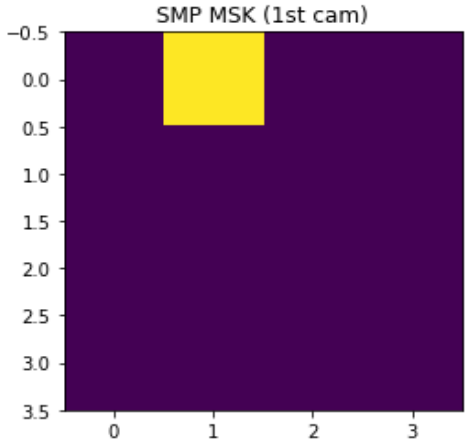
Performance

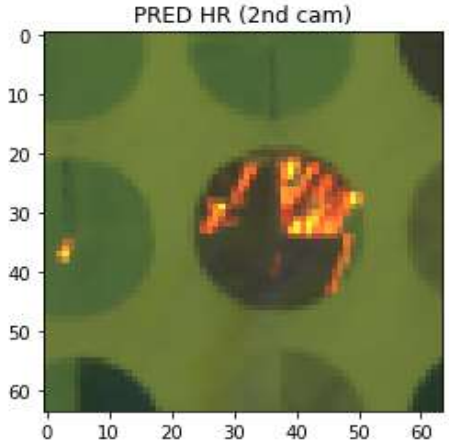


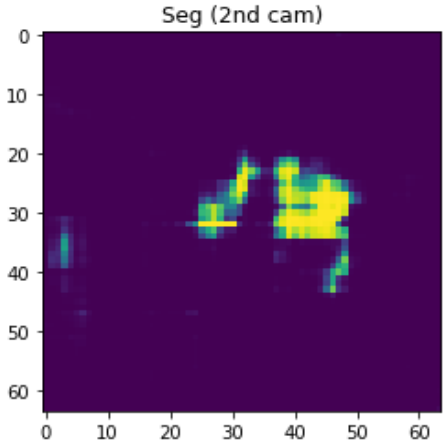
Performance

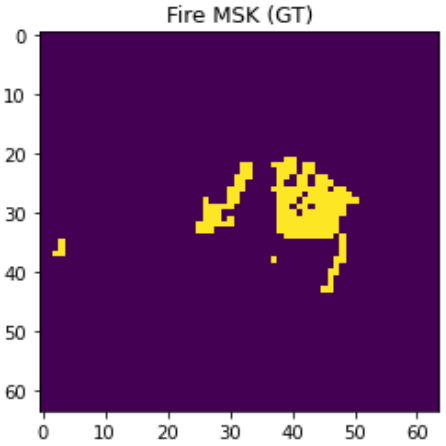




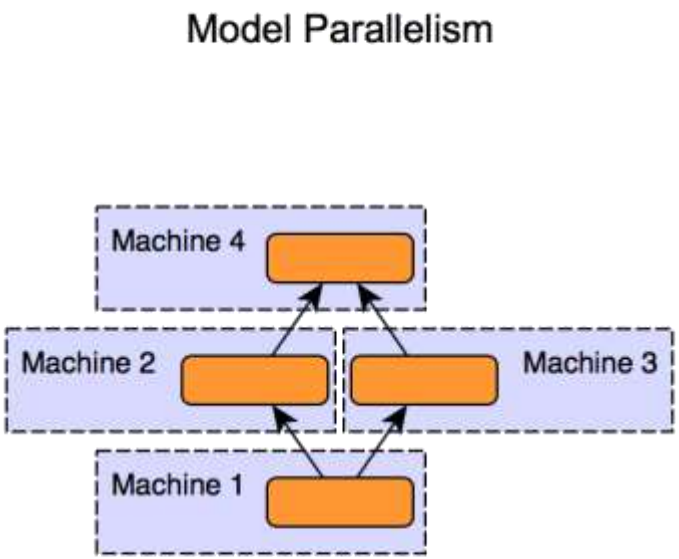


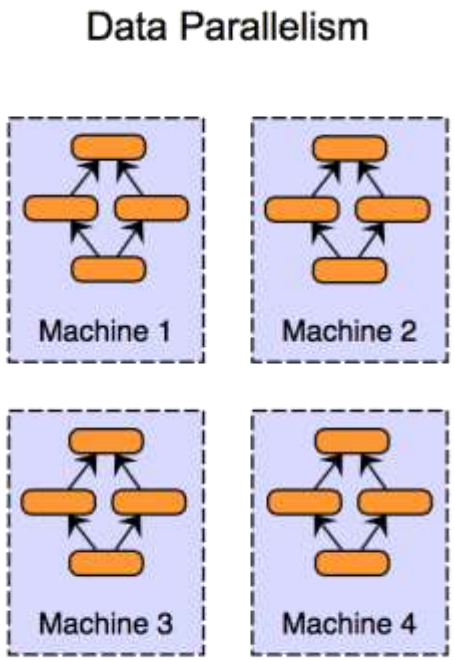






Model vs Data parallelism

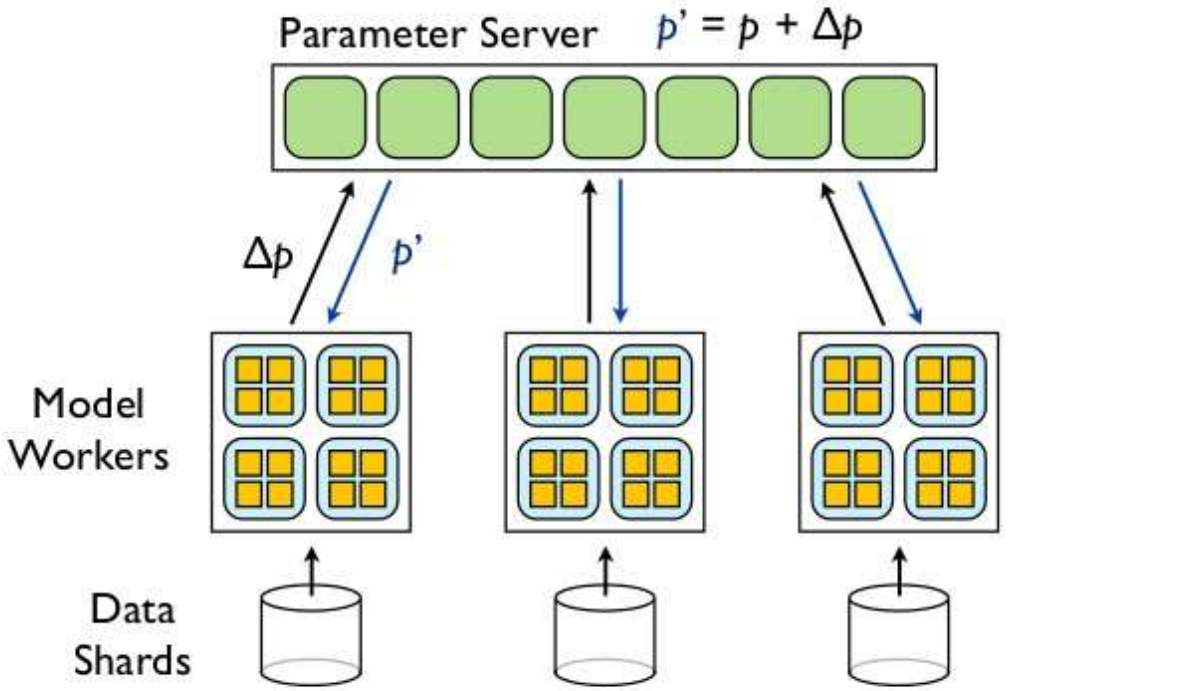




Parameter server approach

Data Parallelism:

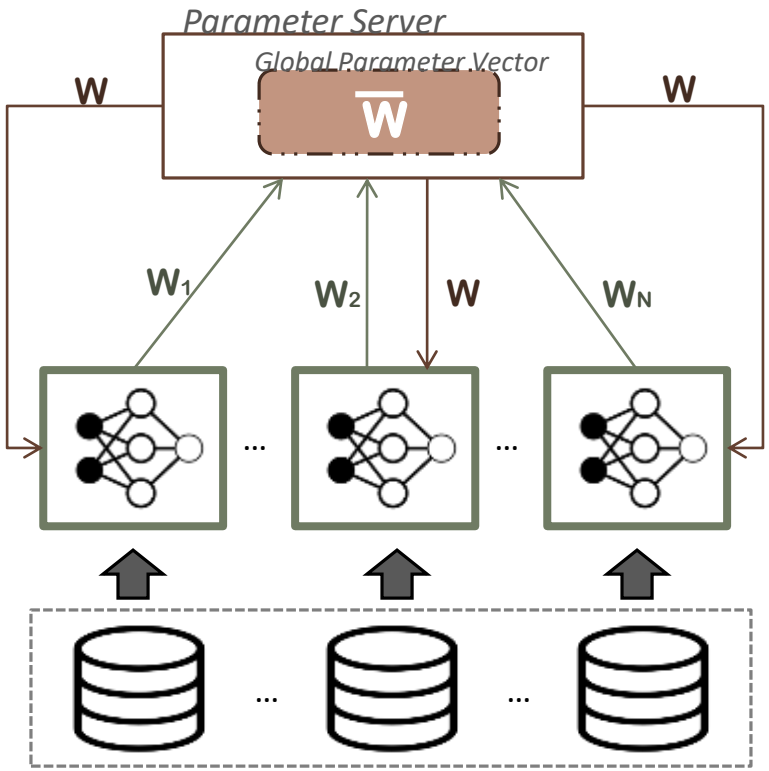
Asynchronous Distributed Stochastic Gradient Descent



Data Parallelism

- Model is replicated over different machines.
- Each worker gets a different shard of the dataset.
- Every update is reported to Parameter Server.

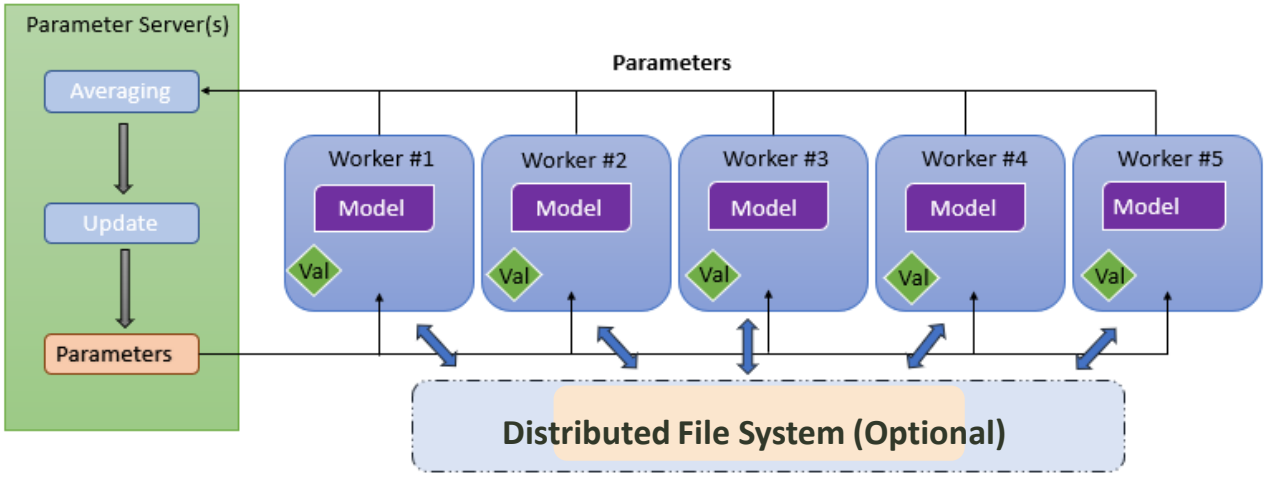
Two approaches:
- *Synchronous*
- *Asynchronous*



Data Parallelism

Synchronous Training

- Parameters are updated on the same timestamp.
- Batch size is multiplied by the number of replicas.
- Very slow: Faster workers wait slower ones.
- Credibility: Parameters are always up to date.

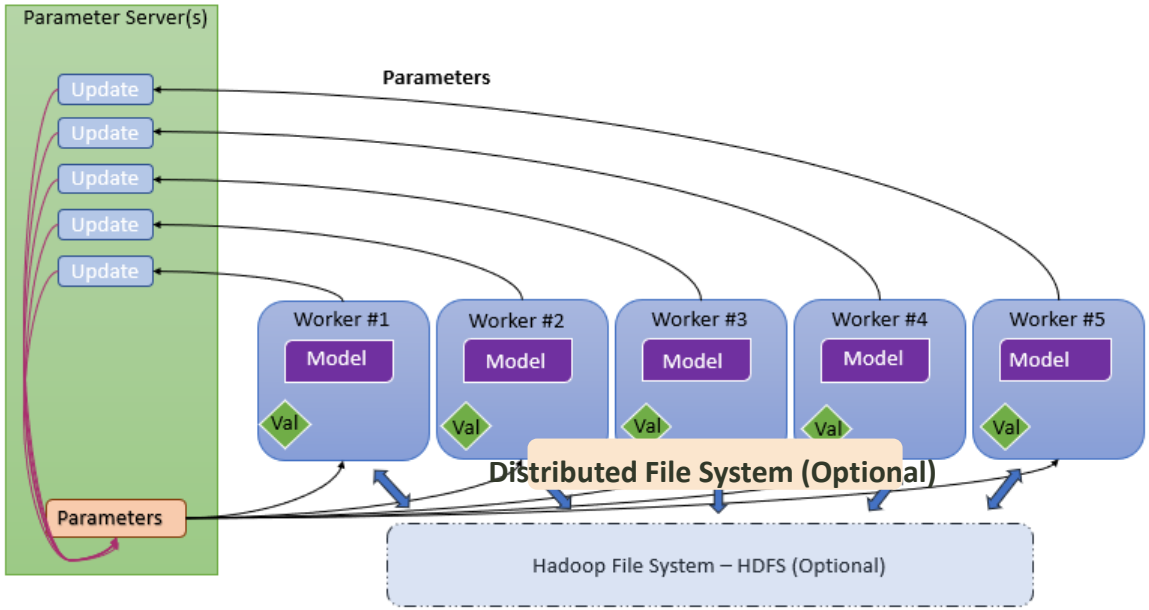


Data Parallelism

- Machines spend more time performing computations, instead of waiting for parameter averaging -> Asynchronous Training

- Higher throughput.

- Stale gradients problem.
  - *When a worker has finished an epoch, parameters may have been updated multiple times.*

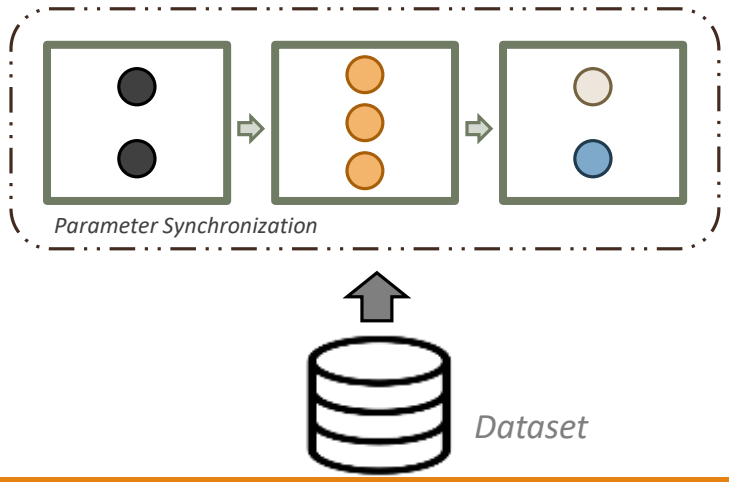


Model Parallelism

- Model is split among different machines.

- Each worker must have access to the whole dataset.
  - *Batch has to be copied to all machines.*

- Parameters are constantly updated.
  - *Interlayer Dependencies.*
  - *Backpropagation requires all-to-all communication on every layer.*

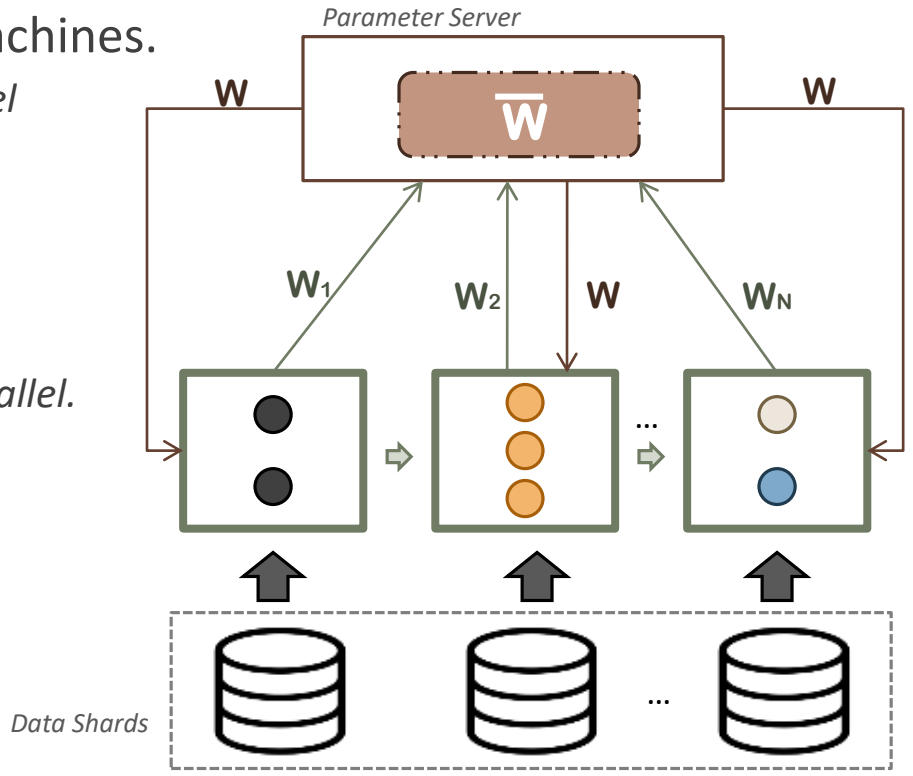


Hybrid Parallelism

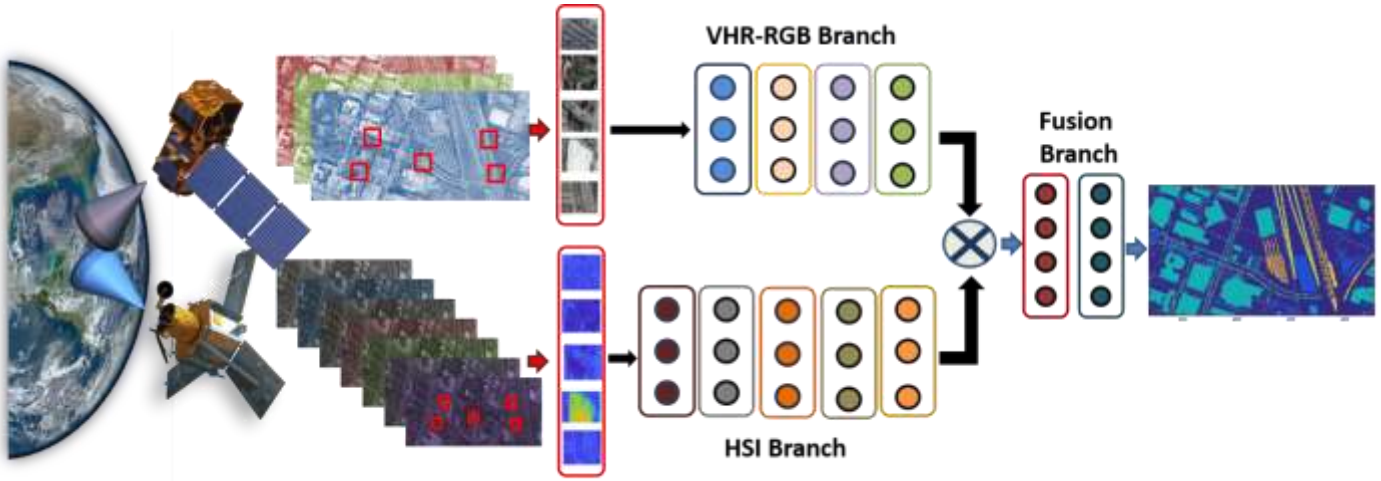
- Layers can be distributed across machines.

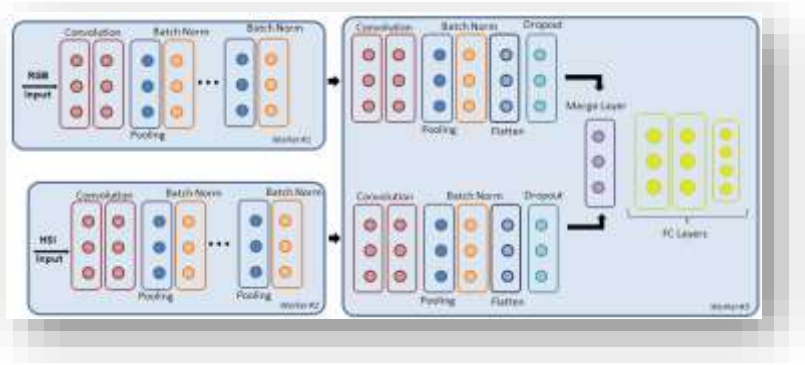
- Data can be distributed across machines.
  - *Enables arbitrary combination of model and data parallelism.*

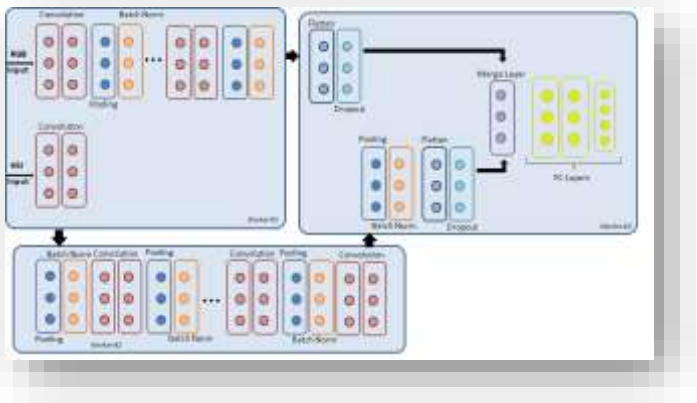
- Often specific to layers types.
  - *Distribute Fully Connected Layers but handle Convolutional Layers data parallel.*



Distributed DNN training



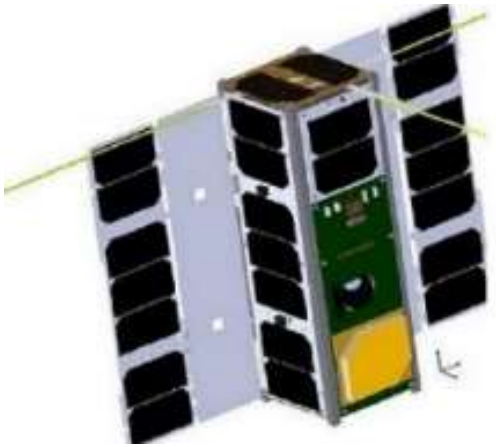




Aspri, M., Tsagkatakis, G., & Tsakalides, P. (2020). Distributed training and inference of deep learning models for multi-modal land cover classification. *Remote Sensing*, *12*(17), 2670.

ESA OPS-SAT

- **OPS-SAT** is a Cubesat platform, currently in operation, developed and managed by ESA.
- Explore the potential of on-board processing
- This capability has been explored for cloud segmentation in [[1]].



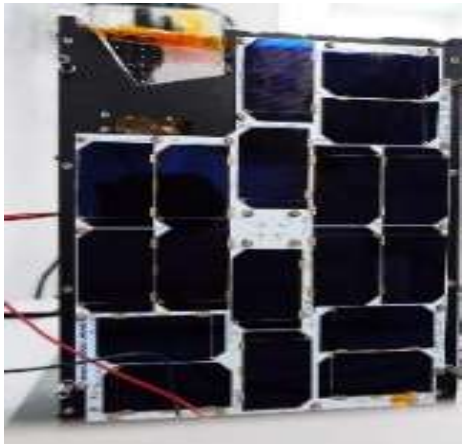
Resolution (GSD)	53m @ 600 km
Swath	~100 km
Detector elements	2048 x 1944
Deployment year	2021

[1] Férésin, Frédéric, et al. "In space image processing using ai embedded on system on module: example of ops-sat cloud segmentation." 2nd European Workshop on On-Board Data Processing. 2021.

Phi-SAT-1 (Φ-Sat-1)

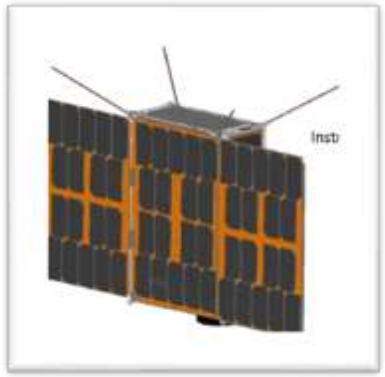
Sensor: Hyperspectral/TIR optical payload - HyperScout®-2 (45+3 bands)

Execution: Intel Movidius board with a Myriad II chip (VPU).



Resolution (GSD)	**83m @ 590km**
Swath	327 x 165 km
Detector elements	4096
Deployment year	2021

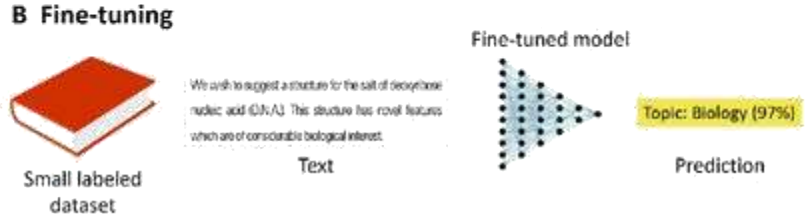
Hardware Realization

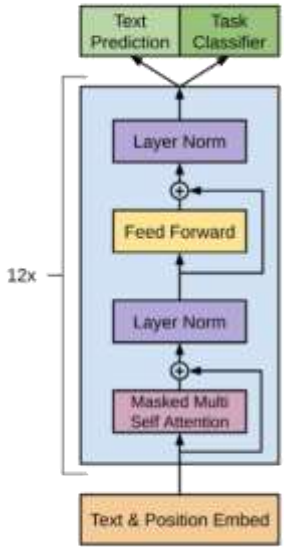


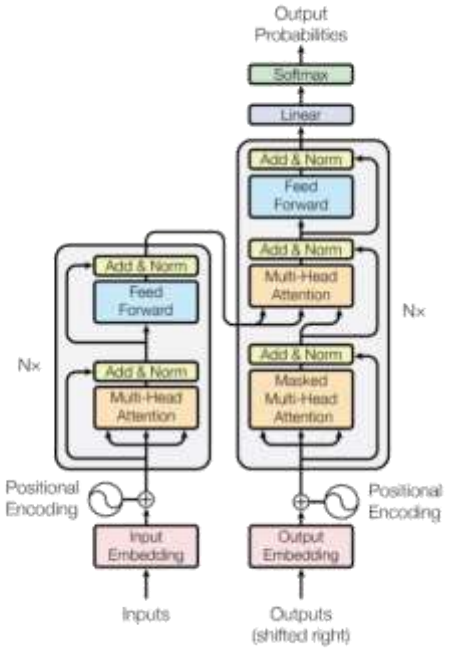
	FPGA Architecture		Intel i-7 7700HQ	Nvidia K2200
	ZCU-102	QFDB		
Clock Frequency (MHz)	250	250	3800	1124
Throughput (Spectra/s)	1084	**4334**	3.47	2000
Latency (s)	0.003	**0.003**	7.6	0.06
GFLOPS	66.1	**265**	0.21	122.5
TDP (Watt)	11.8	47.3	100	300
Energy Consumption (Joule)*	108.8	109.1	288K	1500
Spectra/Joule	91.9	**91.6**	0.035	6.66

Pitsis, George, et al. "Efficient convolutional neural network weight compression for space data classification on multi-fpga platforms." *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019.

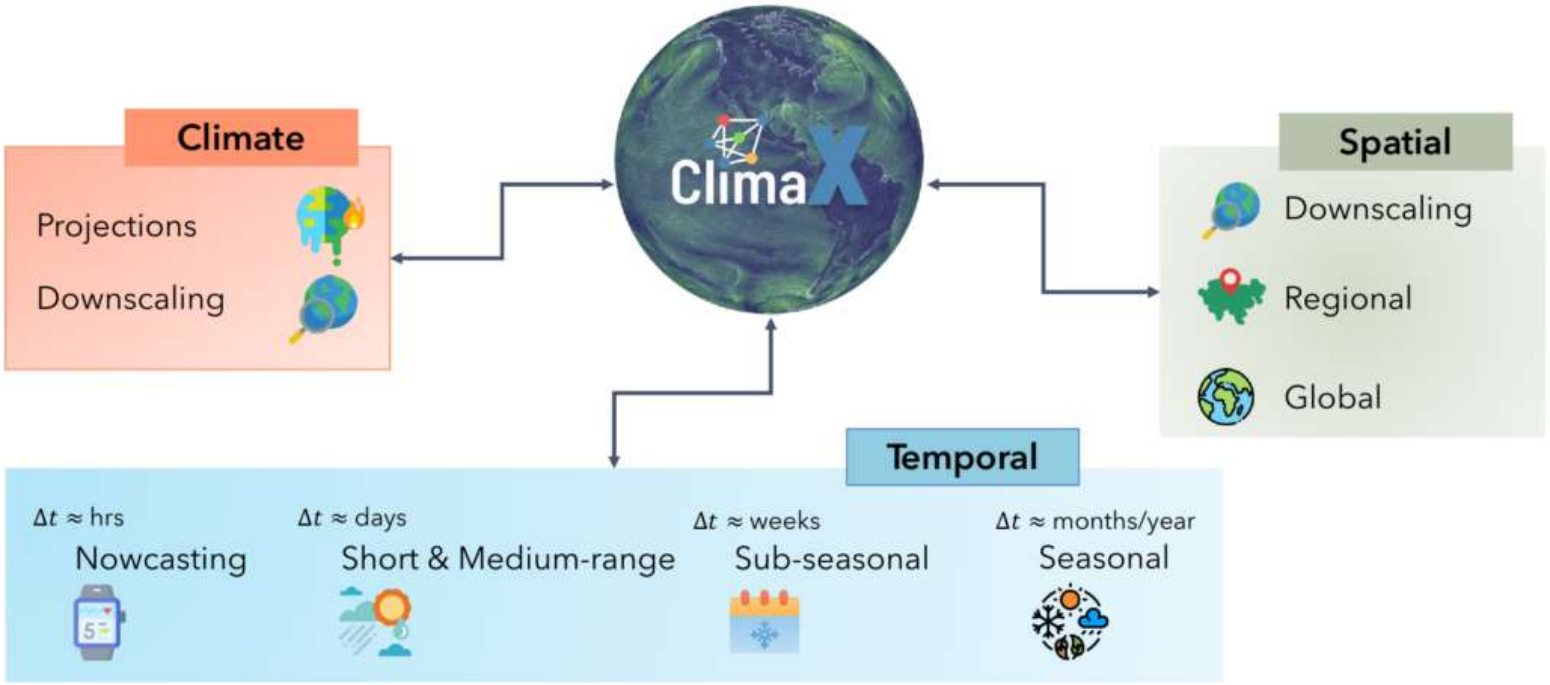
Foundational models in LLM





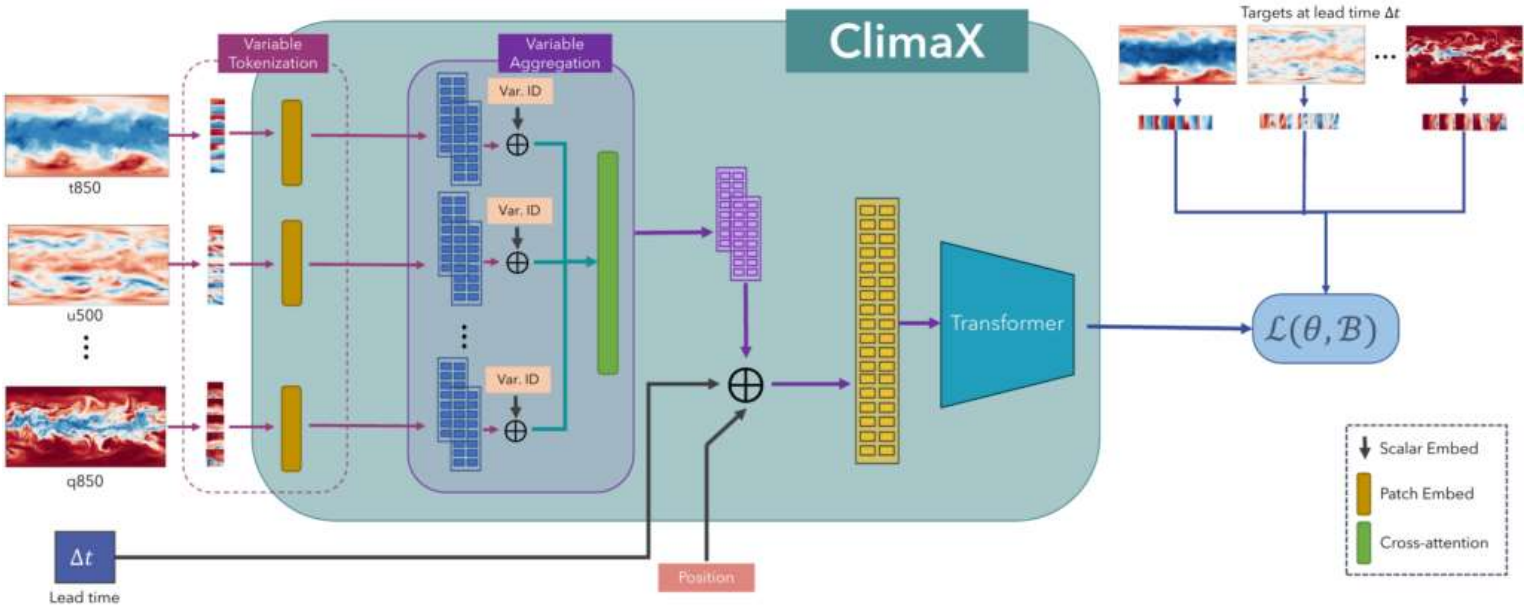


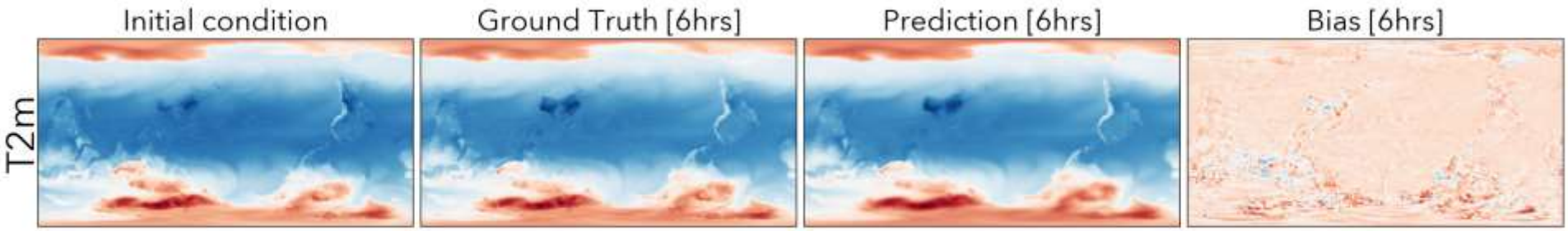
Foundational models in RS



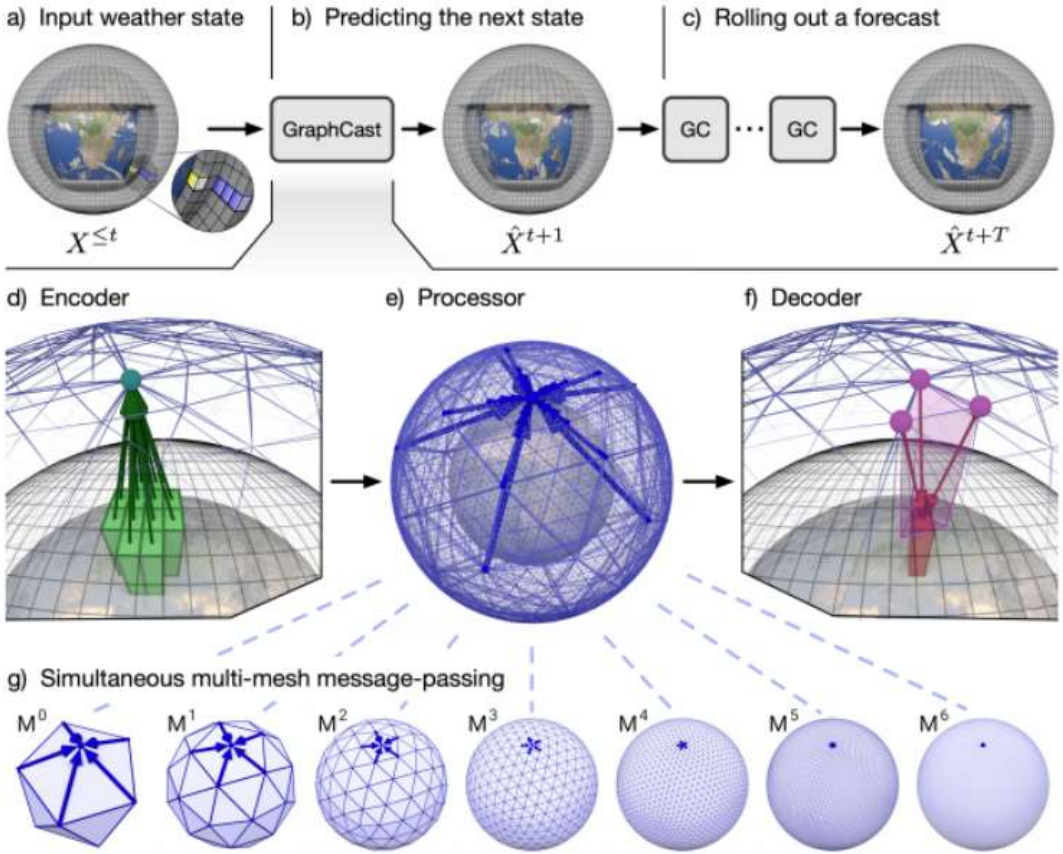
Nguyen, Tung, et al. "ClimaX: A foundation model for weather and climate." *arXiv preprint arXiv:2301.10343* (2023).

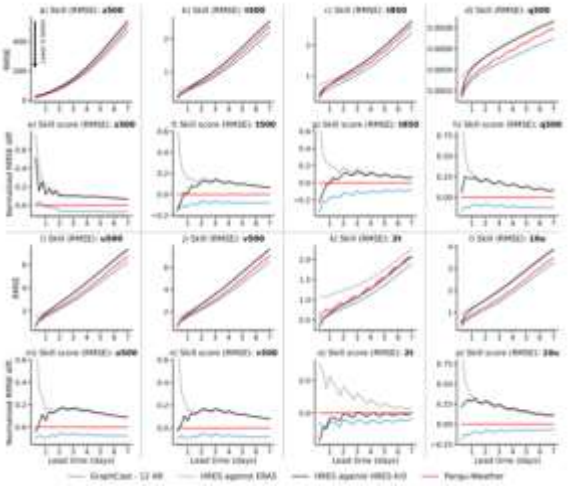
ClimaX


ClimaX
Variable Tokenization
Variable Aggregation
Var. ID
Var. ID
Var. ID
t850
u500
q850
Δt
Lead time
Position
Transformer
Targets at lead time Δt
$\mathcal{L}(\theta, \mathcal{B})$
Scalar Embed
Patch Embed
Cross-attention


Initial condition
Ground Truth [6hrs]
Prediction [6hrs]
Bias [6hrs]
T2m

GraphCast





Lam, Remi, et al. "GraphCast: Learning skillful medium-range global weather forecasting." *arXiv preprint arXiv:2212.12794* (2022).

Next time

Object counting

Regression

Crop classification

Crop yield

Pansharpening

Data fusion

Few-shot learning

Weakly & semi-supervised learning

Active learning

Thank you