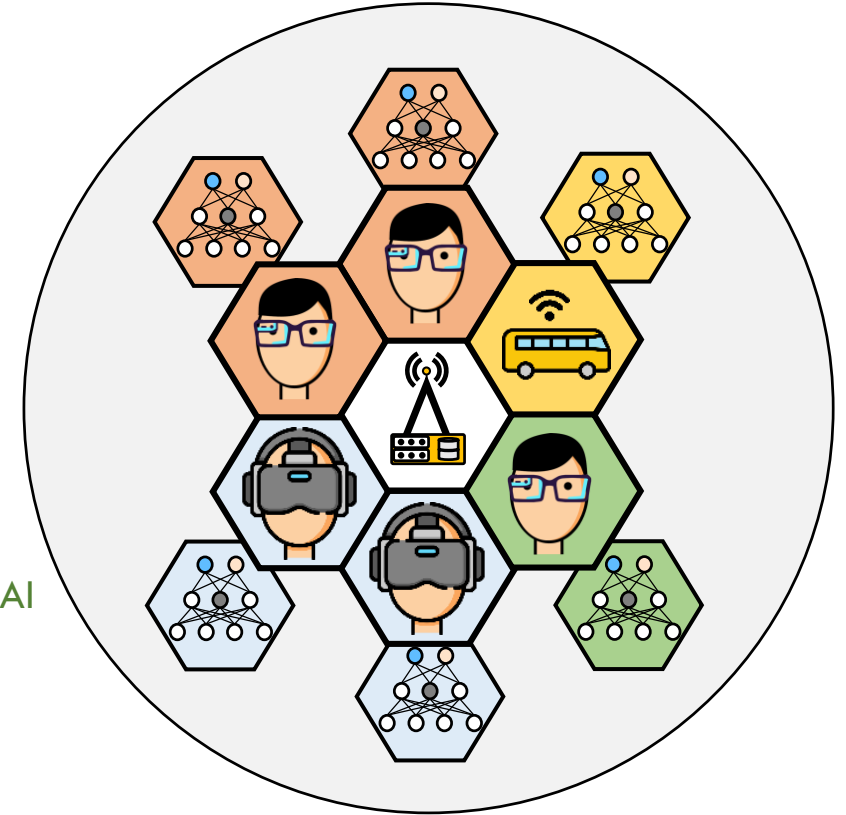


Machine Learning based Edge Computing

Choong Seon HONG

Professor, Department of Computer Science
and Engineering, Kyung Hee University,

- Introduction
 - Edge Computing Technology
 - Multi-access Edge Computing
 - Edge Computing Market Trends
 - AI Applications
 - AI-based Edge Computing
 - Edge AI
 - Technology Trends
- Edge Applications
 - Shopping with Augmented Reality
 - Smart-X Services
 - Image Analytics
- Enabling Technologies to Implement Edge AI
 - Cloud VS. Edge AI
 - Machine Learning Taxonomy - Type of Machine Learning Schemes for Edge AI
 - Supervised learning
 - Unsupervised Learning
 - Reinforcement Learning
- User Cases
 - Use Case-1 : Content Caching at Edge
 - Use Case-2 : Content Caching at Virtualized Edge
 - Use Case-3 : License Plate Detection
- Conclusions

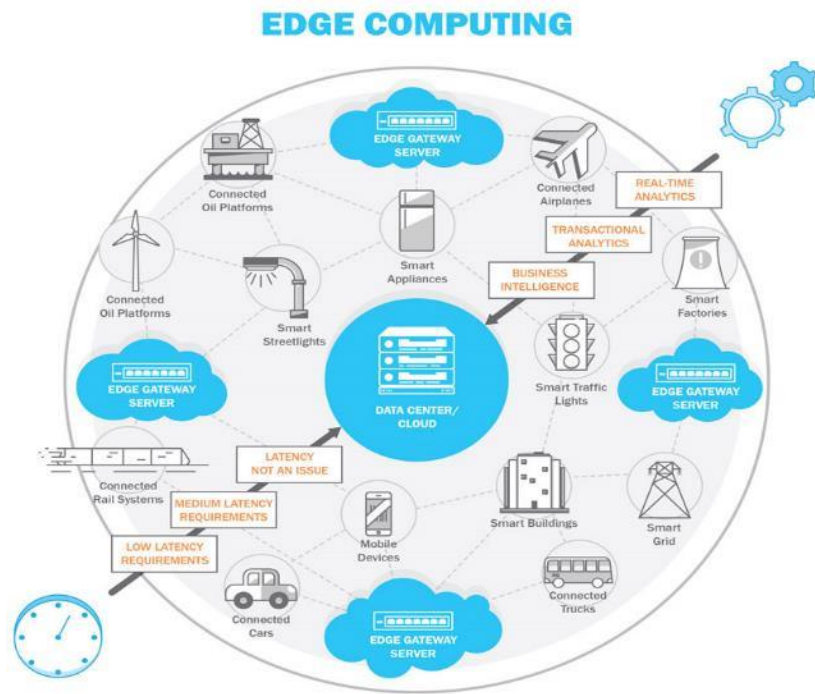


Introduction

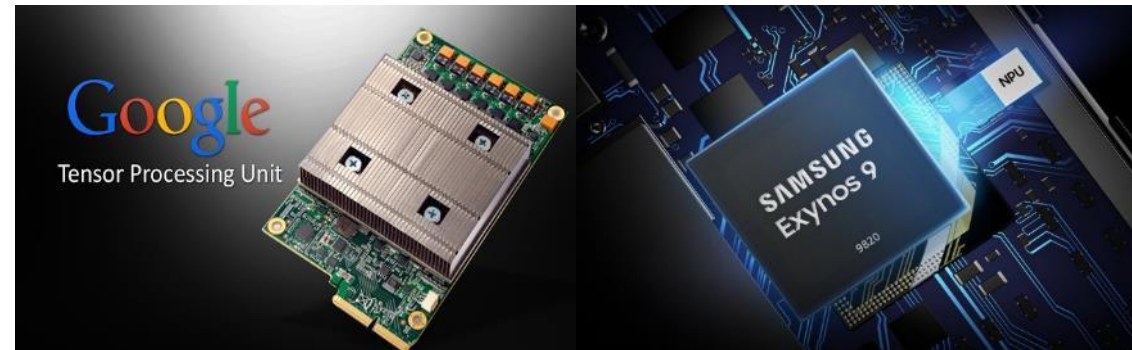
- Edge Computing Technology
- Multi-access Edge Computing
- Edge Computing Market Trends
- AI Applications
- AI-based Edge Computing
- Edge AI
- Technology Trends

✓ Changing the computing paradigm (Centralized Cloud Computing → Distributed Edge Computing)

- Increasing services that require real-time, large capacity, and low latency
- Edge computing is expected to be used in industrial/entertainment fields such as AR/VR, unmanned aerial vehicles, autonomous vehicles, and smart factories.
- In order to accommodate edge computing technology, the system semiconductor industry is focusing on NPU development.



Edge Computing



Google TPU

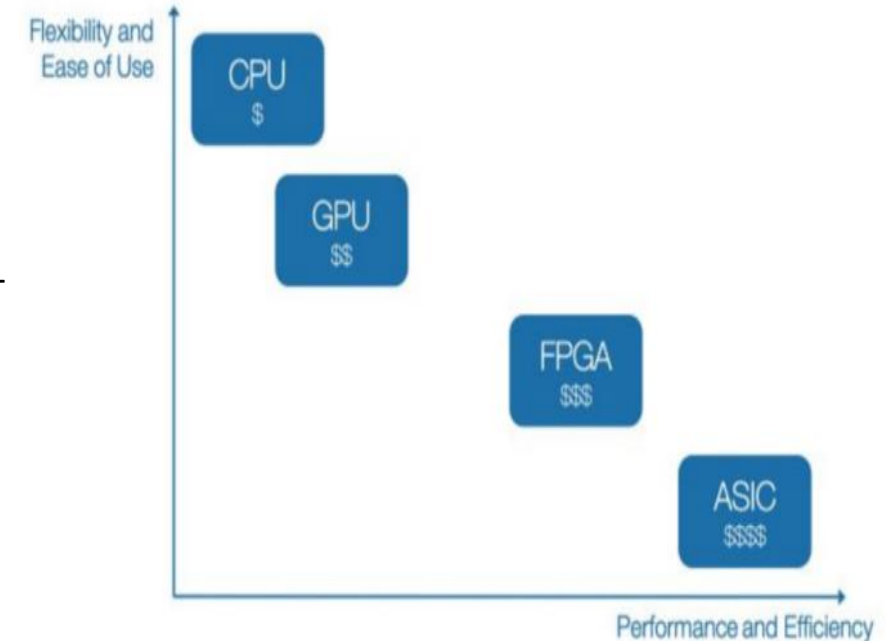
Samsung NPU

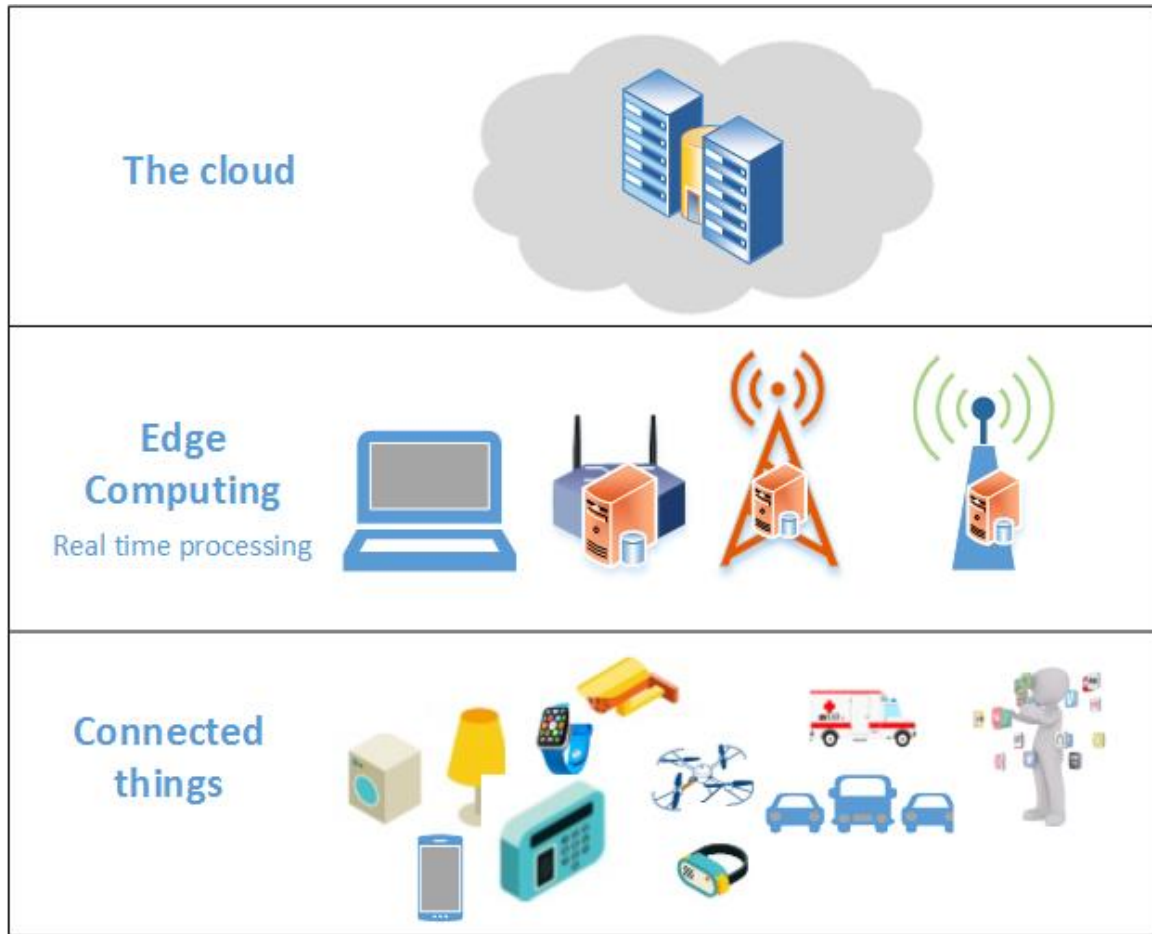


APPLE Neural Engine

NVIDIA Jetson Xavier

- **Central Processing Unit (CPU)**
 - Cheap, flexible, slow
- **Graphical Processing Unit (GPU)**
 - Expensive, high throughput, great for batching to utilize parallel processing.
- **Field-Programmable Gate Array (FPGA)**
 - Expensive, fast, low power, reprogrammable for custom solutions.
- **Application-Specific Integrated Circuit (ASIC)**
 - Extremely expensive to design but inexpensive when build for scale, custom-made chip.
- **Tensor Processing Unit (TPU)**
 - ASIC specializing in operations for neural networks.
- **Edge TPU**
 - Smaller than a US penny, accelerates inference on the edge.
- **Neural Processing Unit (NPU)**
 - Often used by smartphone manufactures and it is a dedicated chip for accelerating neural network inference.



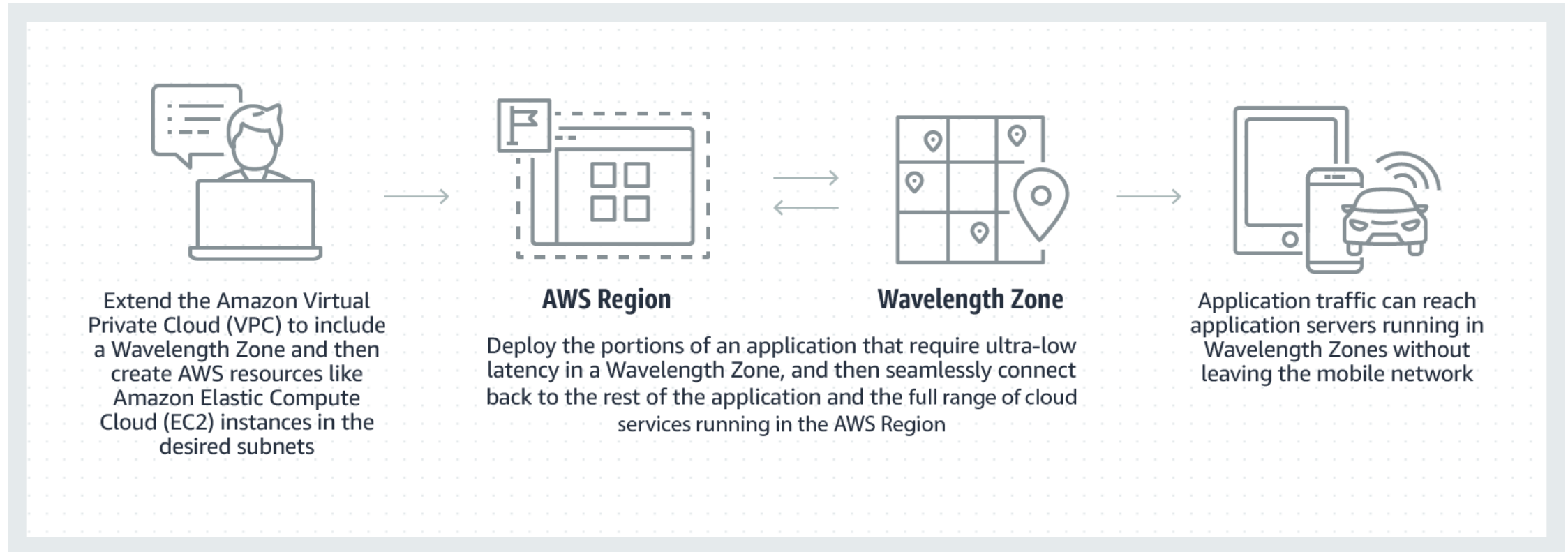


- Industry investment and research interest in edge computing, in which computing and storage nodes are placed at the Internet's edge in close proximity to the things (mobile devices, sensors, etc.), have grown dramatically in recent years [1]
- In 2013, **Nokia and IBM** jointly introduced the **Radio Applications Cloud Server (RACS)**, an edge computing platform for 4G/LTE networks
- In September 2014, a **mobile edge computing** standardization effort began under the auspices of the **European Telecommunications Standards Institute (ETSI)**
- In June 2015, the **Open Edge Computing (OEC)** was introduced by **Vodafone, Intel, and Huawei** in partnership with **Carnegie Mellon University (CMU)**, in which one year later **Verizon, Deutsche Telekom, T-Mobile, Nokia, and Crown Castle** were included
- In 2020, **KT Corp.** formed an alliance with global telecom operators, including U.S.-based **Verizon Wireless**, to develop global specifications and standards for 5G MEC interoperability[*].
- On July 2020, Major mobile carrier **SK Telecom Co.** joined hands with the country's largest food delivery operator **Woowa Brothers Corp** to develop an autonomous robot delivery service using MEC[*].
- On Sept 2020, **LG Uplus Corp**, a major South Korean mobile carrier joining forces with **Google** to jointly develop 5G mobile edge computing (MEC) technology [*].
- On August 6 2020, **Verizon** announced the launch of **5G Edge**, where application developers and business clients can now access the computing and storage resources of **AWS Wavelength** via Verizon's 5G Edge.

1. Satyanarayanan, Mahadev. "The emergence of edge computing." Computer 50.1 (2017): 30-39.

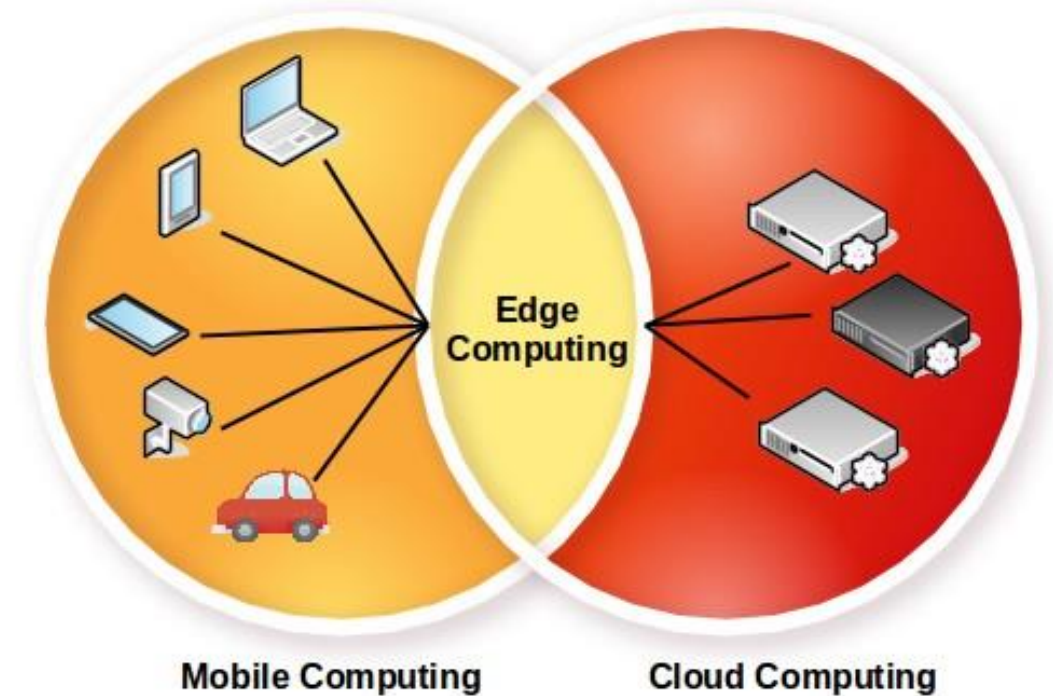
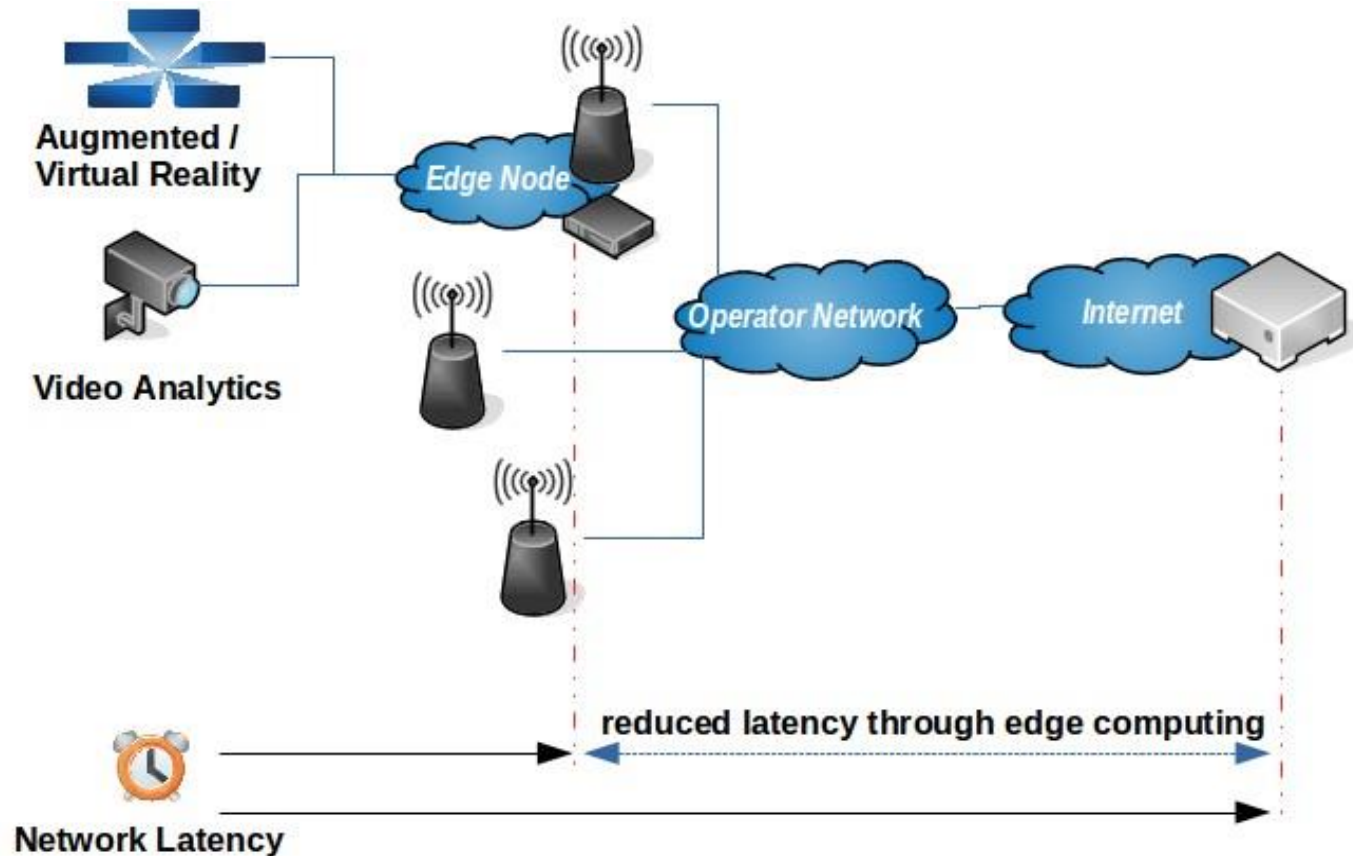
* <https://www.expresscomputer.in/news/lg-uplus-joins-google-cloud-for-5g-mobile-edge-computing-tech/63927/>

- AWS Wavelength



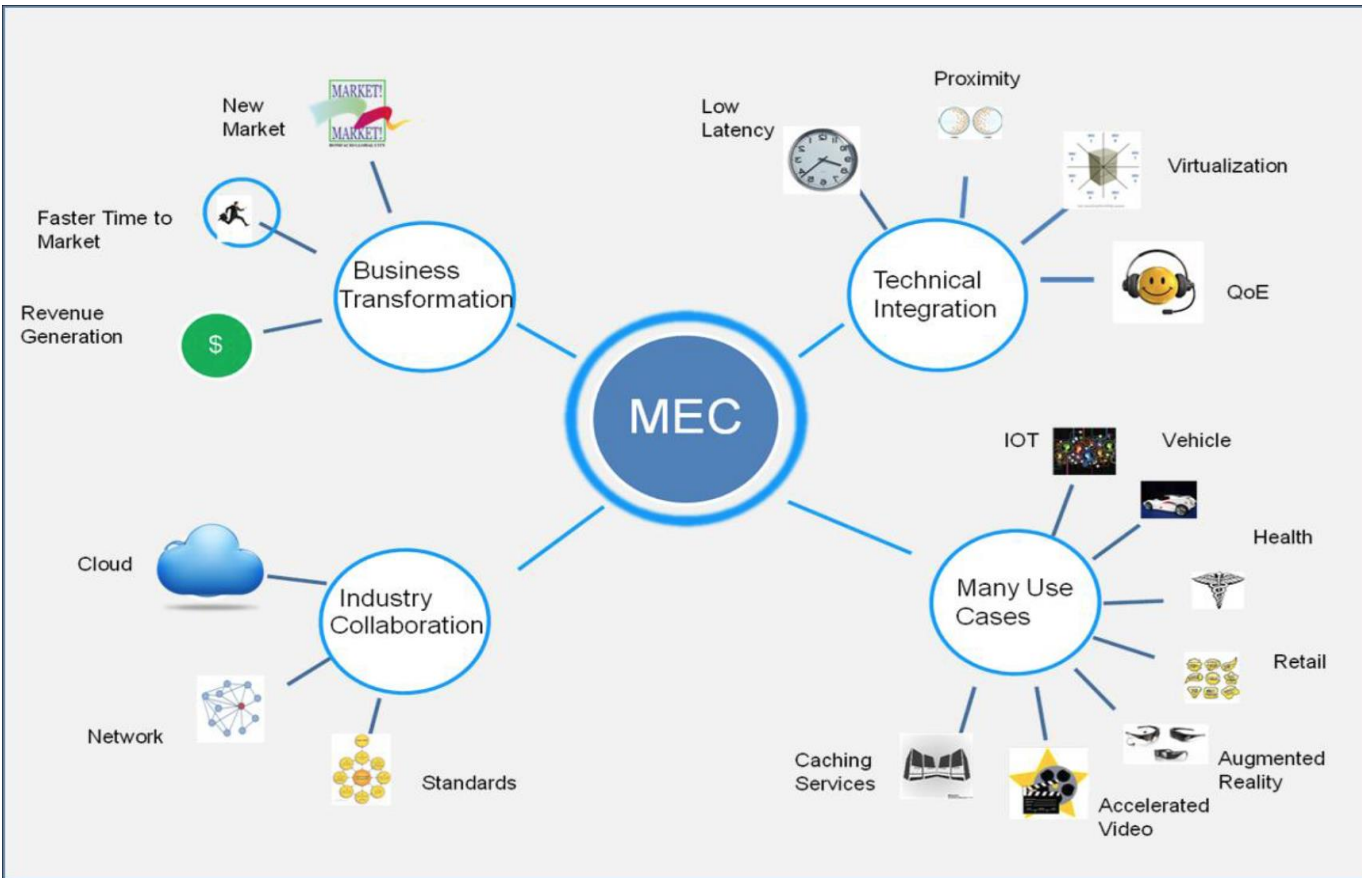
<https://aws.amazon.com/wavelength/>

Vodafone, Intel, and Huawei in partnership with Carnegie Mellon University: Open Edge Computing (OEC)



Source: Rolf Schuster, Prakash Ramchandran "Open Edge Computing - From Vision To Reality", June 2016, OPNFV Design Summit, Berlin, Germany,

<http://openedgecomputing.org>

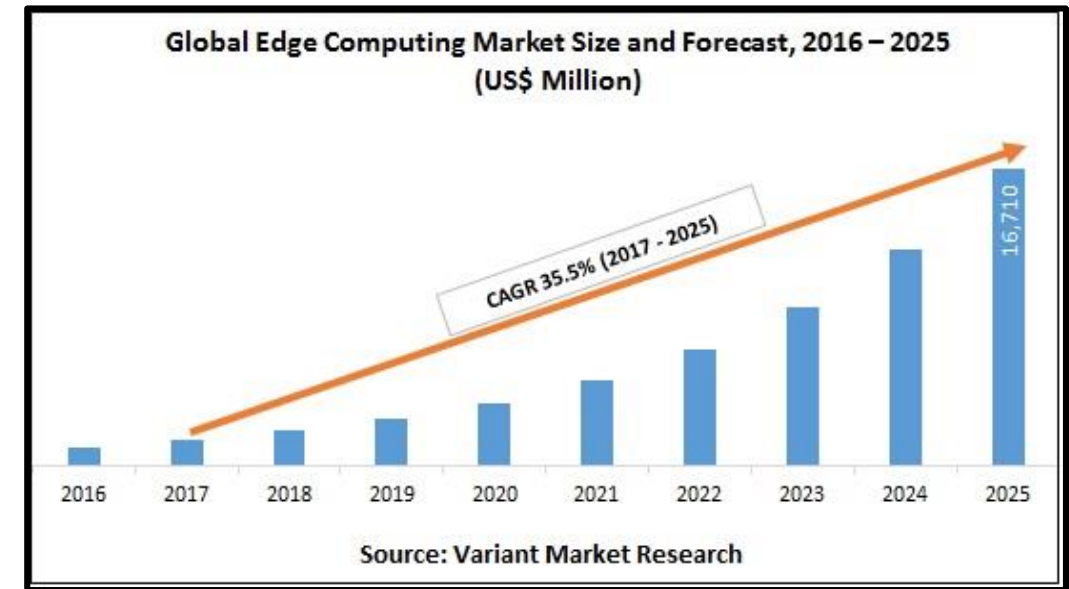
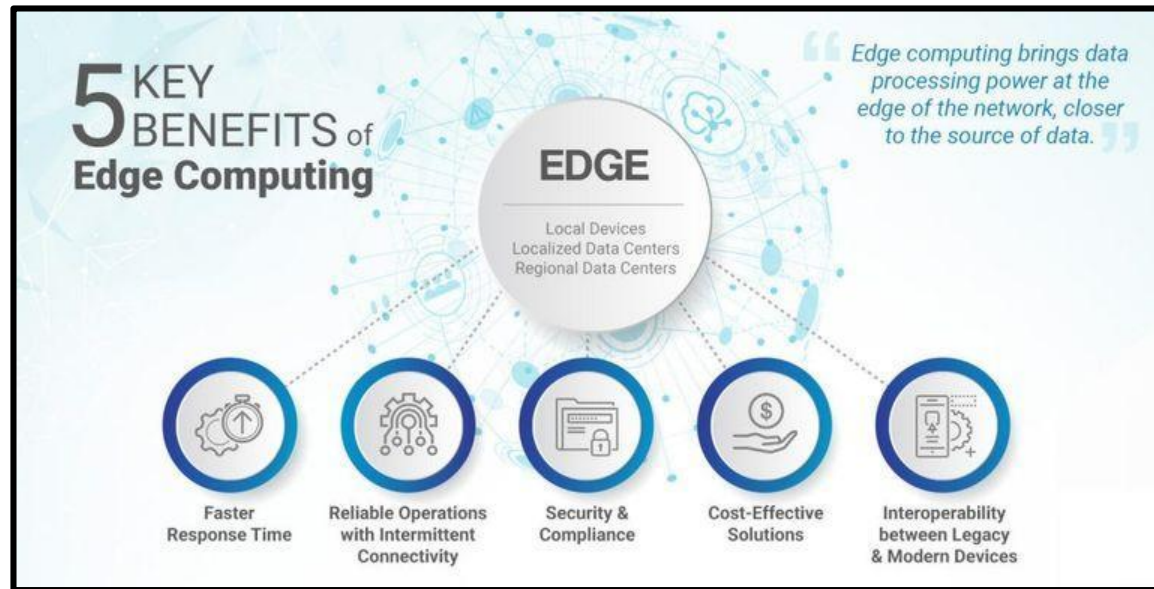


Multi-access Edge Computing (MEC)

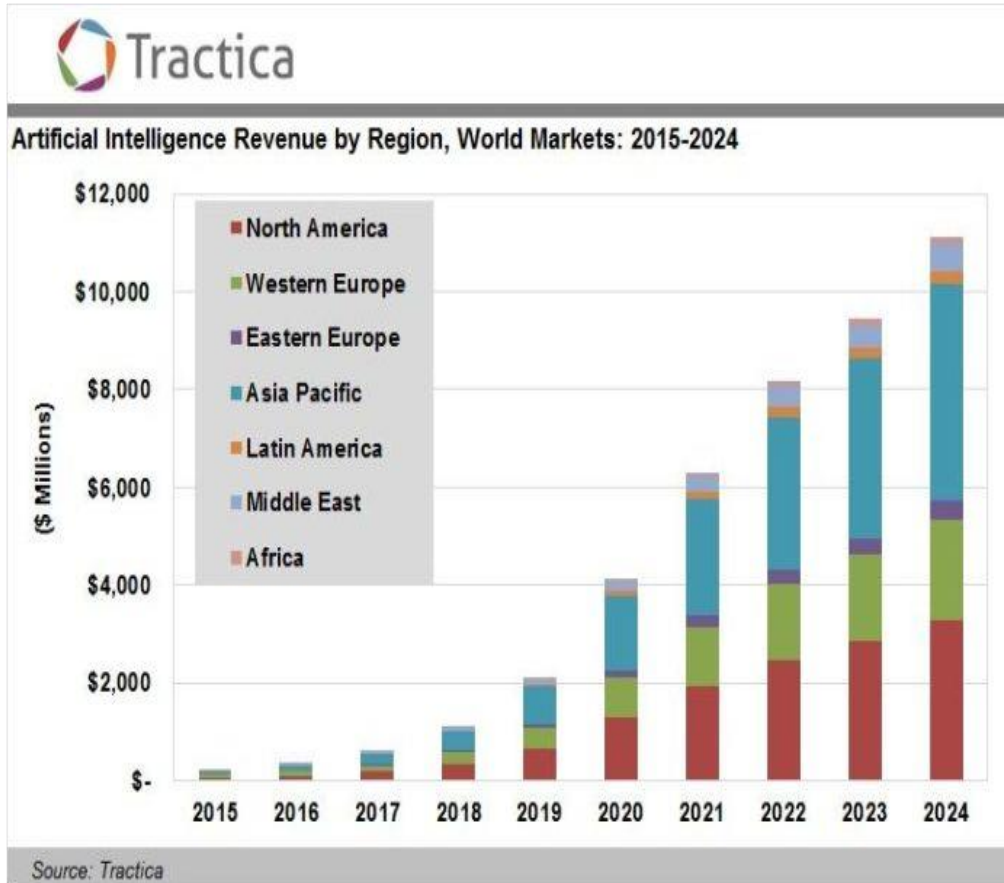
- In 2017, in order to reflect the growing interest in MEC from non-cellular players, ETSI MEC industry standards group changed “mobile edge computing” to “multi-access edge computing” [1]

Benefits of MEC

- **Move cloud infrastructure closer to end users by pushing not only computation, but also caching, communication, and control (4C) to the edge of the network**



- The global edge computing market is expected to reach about \$ 16.71 billion by 2025, (2017~2025) Year average growth rate of 35.5%
- Among the many fields that utilize Edge Computing in addition to 5G, it is expected to show the highest growth rate in areas where fast service provision is important (Smart City, Smart Factory, etc.)



AI technology is currently being applied to solutions for various use cases such as **Agriculture**, **Financial Services**, **Retail**, and **Energy**, and it is expected that sales from AI technology will continue to increase globally.



Agriculture

- To evaluate the best crop choices against various parameters such as soil quality and the needs of the farmers



Financial Services

- Many investment firms are using deep learning algorithms to improve speed, optimize and increase the efficiency of recognizing trends across vast data sets to obtain competitive advantages.



Retail

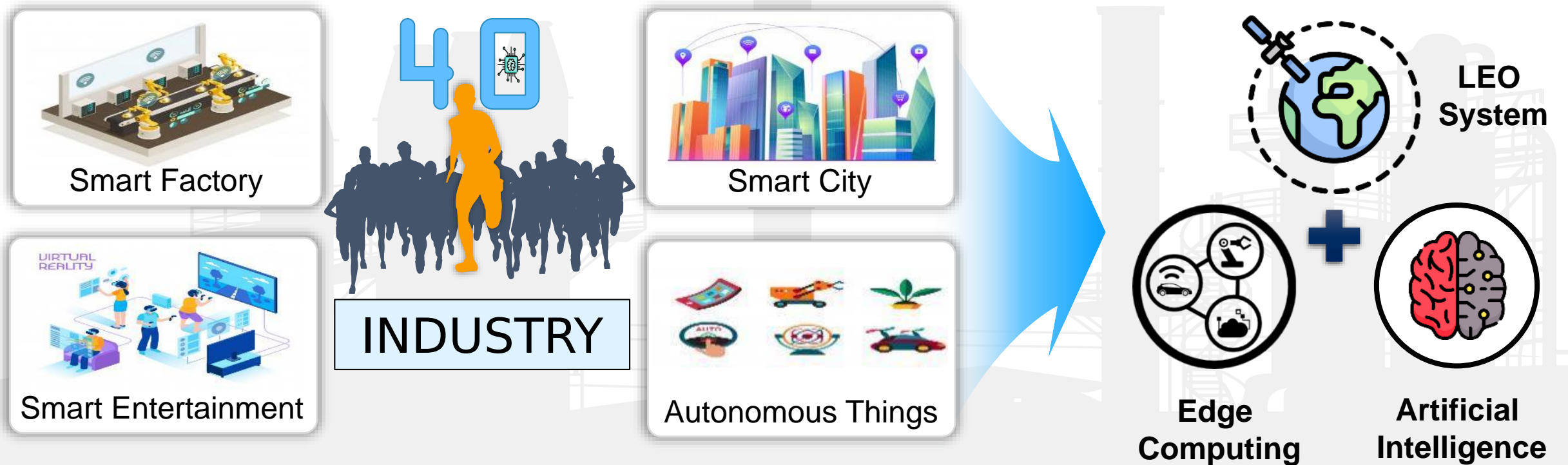
- Optimization of pricing based on the product demand.
- To offer seamless experience to customers and deliver greater customer satisfaction.



Energy

- To improve and increase efficiency and awareness of gas plants and power grid systems.

- ✓ The 4th Industrial Revolution is opening an era of hyper-connectivity and the need to provide services using artificial intelligence is emerging.
 - ✓ By applying AI technology to the edge, it is possible to draw fast and accurate results for analysis and decision making in IoT environment
- **Increasing the role of artificial intelligence algorithms in edge computing environments**



The current premise in classical ML is based on a single node in a centralized and remote data center with full access to a global dataset and a massive amount of storage and computing power, sifting through this data for inference.

Nevertheless the advent of a new breed of intelligent devices and high-stake applications ranging from **drones** to augmented/virtual reality (**AR/VR**) applications, and **self driving vehicles**, makes cloud-based ML inadequate. These applications are real-time, cannot afford latency, and must operate under high reliability, **even when network connectivity is lost**.

WHY AI AT THE EDGE MATTERS

Bandwidth



1 billion cameras WW (2020)
30B Inference/Second

Latency



30 images per second
200ms latency

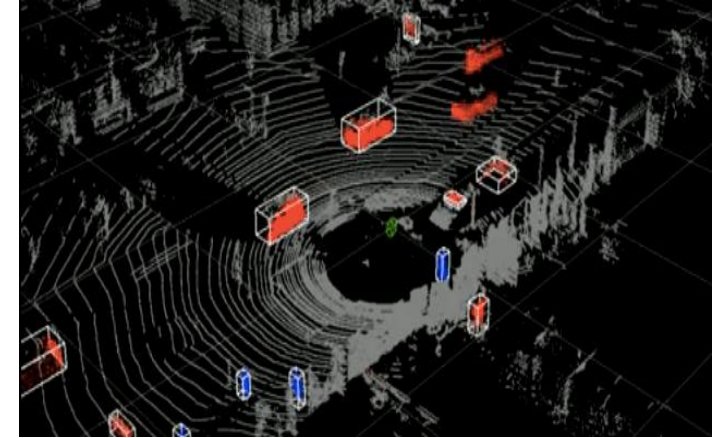
Availability



50% of world at less than 8mbps
Only 73% 3G/4G availability WW

"Billions of intelligent devices will take advantage of DNNs to provide personalization and localization as GPUs become faster and faster over the next several years."
Tractica

- Indeed, an autonomous vehicle that needs to apply its brakes, cannot allow even a millisecond of latency that might result from cloud processing, requiring split second decisions for safe operation [2], [3].
- A user enjoying visual-haptic perceptions requires not only minimal individual perception delays but also minimal delay variance, to avoid motion sickness [4], [5].
- A remotely controlled drone or a robotic assembler in a smart factory should always be operational even when network connection is temporarily unavailable [6]–[8], by sensing and reacting rapidly to local (and possibly hazardous) environments.



[2] S.-C. Lin, Y. Zhang, C.-H. Hsu, M. Skach, M. E. Haque, L. Tang, and J. Mars, “The architectural implications of autonomous driving: Constraints and acceleration,” in Proc. of the 23rd ACM ASPLOS, ASPLOS ’18, (Williamsburg, VA, USA), pp. 751–766, ACM, Mar. 2018.

[3] M. K. Abdel-Aziz, C.-F. Liu, S. Samarakoon, M. Bennis, and W. Saad, “Ultra-reliable low-latency vehicular networks: Taming the age of information tail,” in Proc. of GLOBECOM [accepted], (Abu Dhabi, UAE), Dec. 2018.

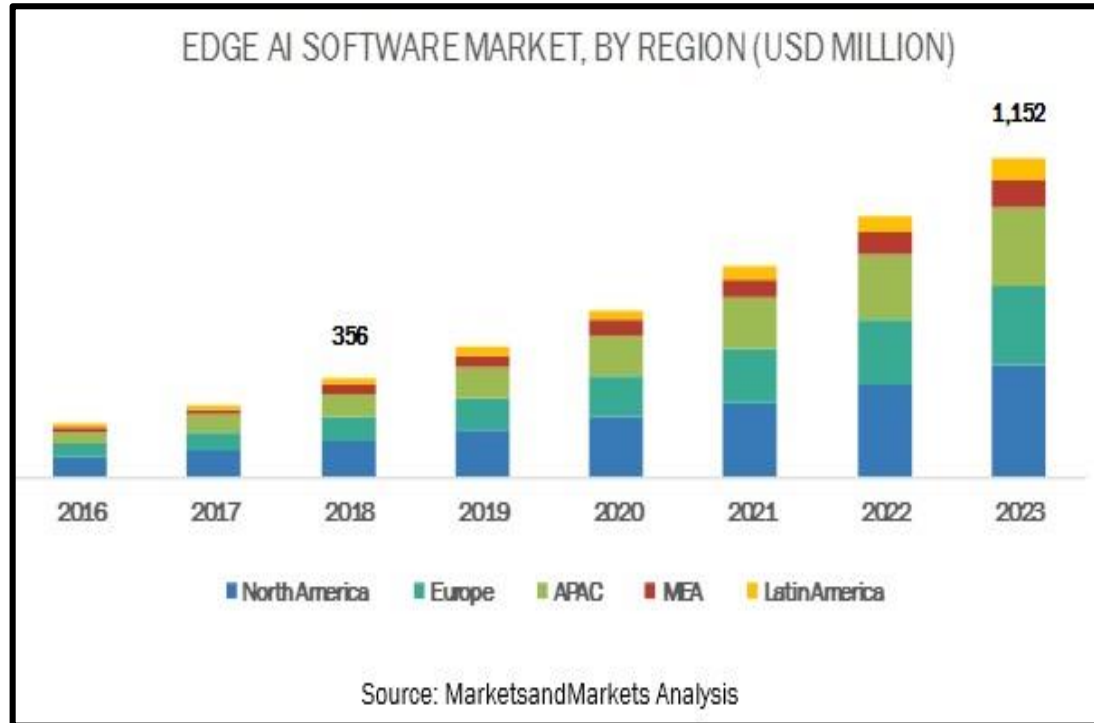
[4] J. Park and M. Bennis, “URLLC-eMBB slicing to support VR multimodal perceptions over wireless cellular systems,” ArXiv preprint, vol. abs/1805.00142, May 2018.

[5] ABI Research and Qualcomm, “Augmented and virtual reality: The first wave of 5g killer apps,” white paper, Feb. 2017.

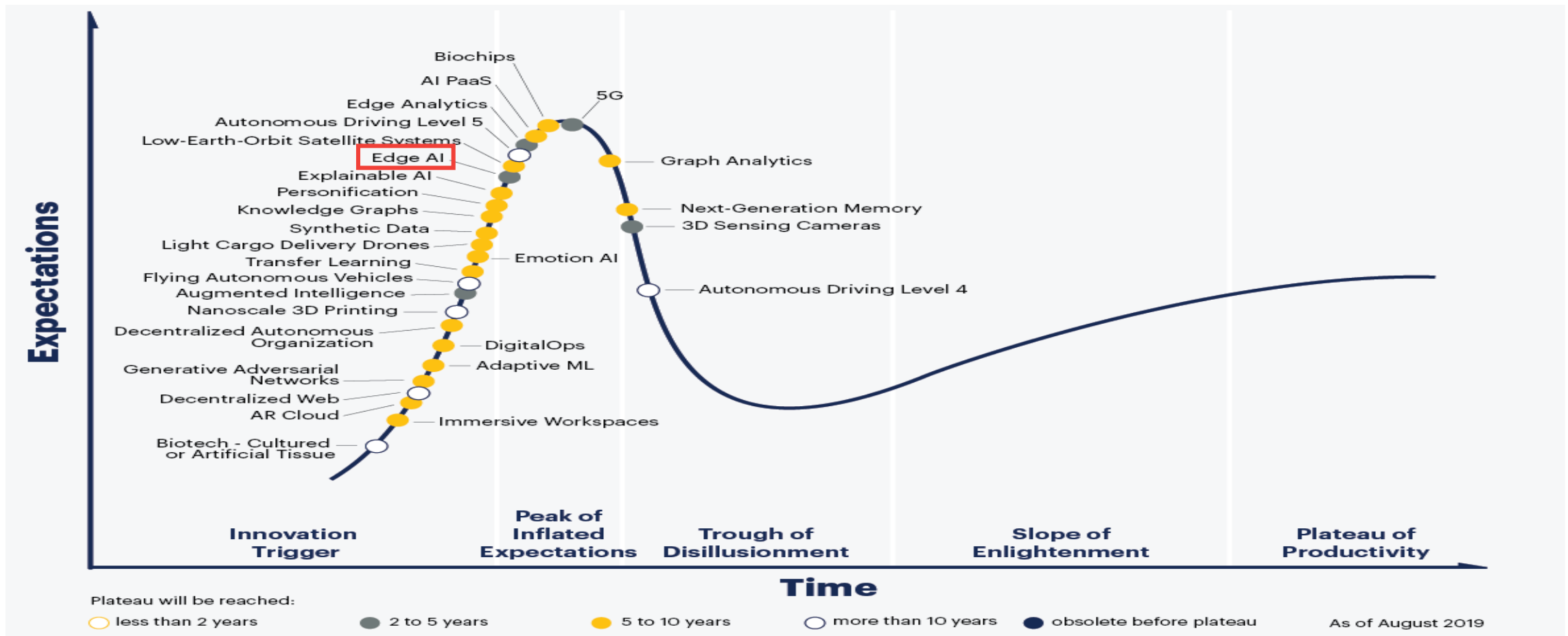
[6] T. Kagawa, F. Ono, L. Shan, K. Takizawa, R. Miura, H. Li, F. Kojima, and S. Kato, “A study on latency-guaranteed multihop wireless communication system for control of robots and drones,” in Proc. of 20th WPMC, (Yogyakarta, Indonesia), pp. 417– 421, Dec. 2017.

[7] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, “Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs,” IEEE Transactions on Wireless Communications, vol. 15, pp. 3949–3963, June 2016.

[8] A. Fotouhi, H. Qiang, M. Ding, M. Hassan, L. Galati Giordano, A. Garcia-Rodriguez, and J. Yuan, “Survey on UAV cellular communications: Practical aspects, standardization advancements, regulation, and security challenges,” ArXiv preprint, vol. abs/1809.01752, Sept. 2018.



- The Edge AI software market is expected to grow at a CAGR of approximately 26.5% from \$ 356 million in 2018 to \$ 1,152 million in 2023, driven by increased cloud loads and accelerated development of various AI applications.

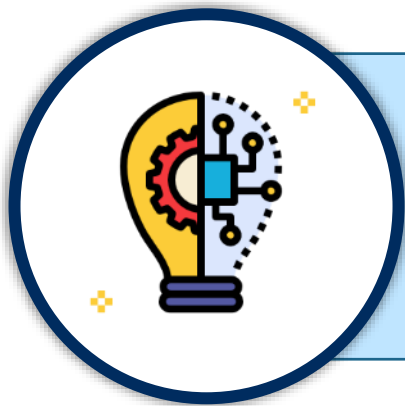


✂ Source : Gartner Hype Cycle for Emerging Technologies 2019

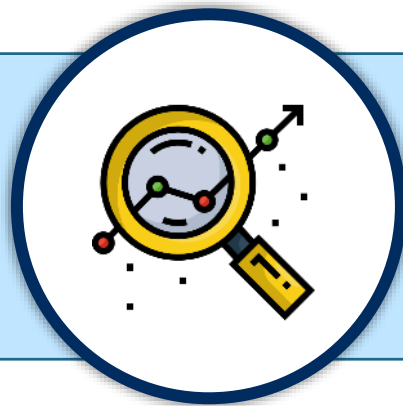
Gartner 2020 Ten Trends in the Future

① Hyperautomation, ② Multiexperience, ③ Democratization of Expertise, ④ Human Augmentation, ⑤ Transparency and Traceability, ⑥ Empowered Edge, ⑦ Distributed Cloud, ⑧ Autonomous Things, ⑨ Quantum Computing, ⑩ AI Security

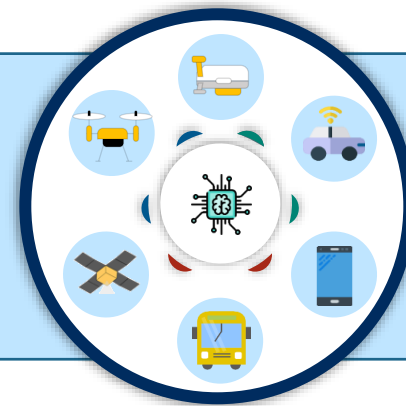
Future technology change



Hyper automation



Democratization of Expertise



Empowered Edge



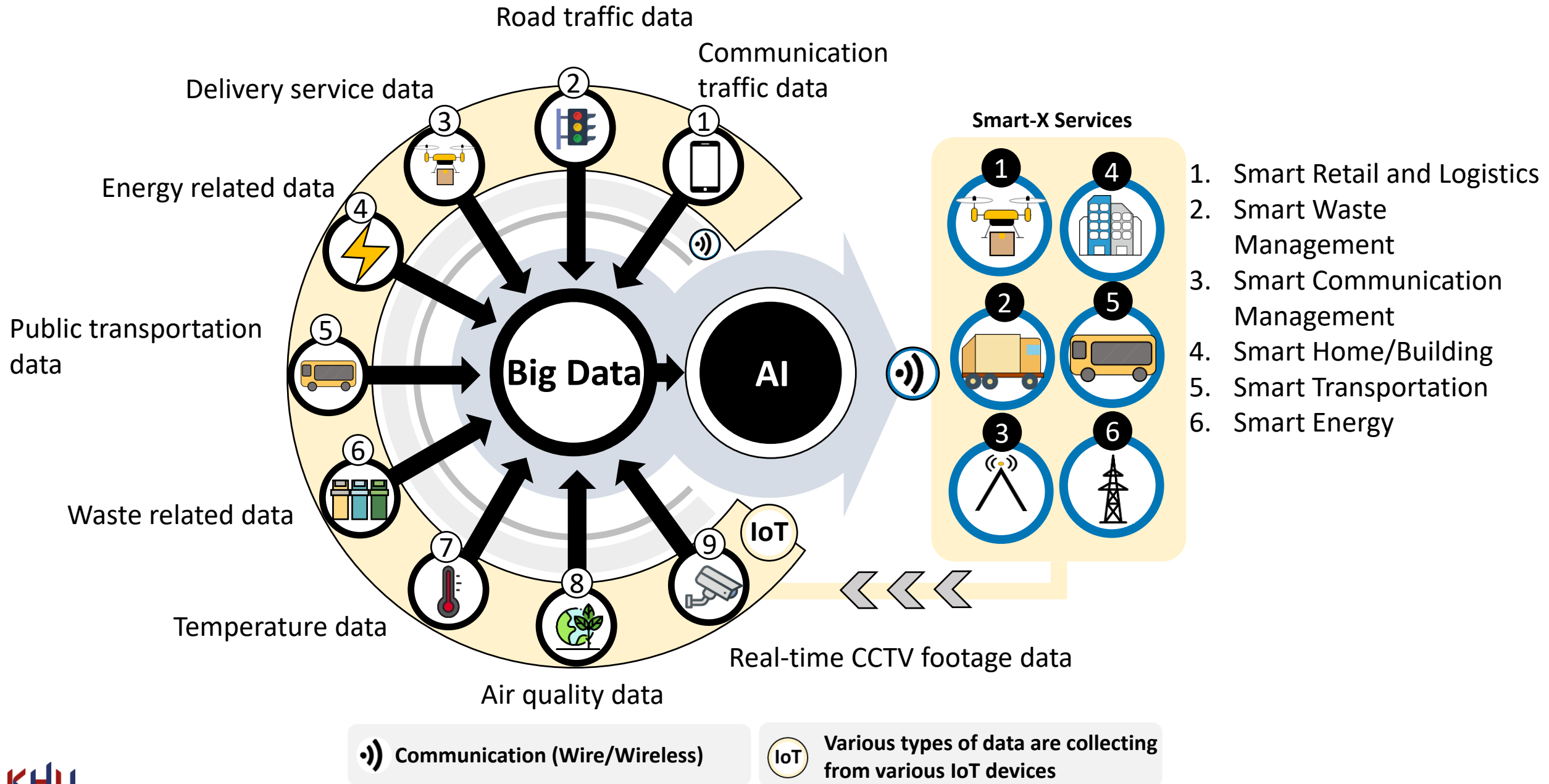
Autonomous Things

※ 출처: Top 10 Strategic Technology Trends 2020, Gartner

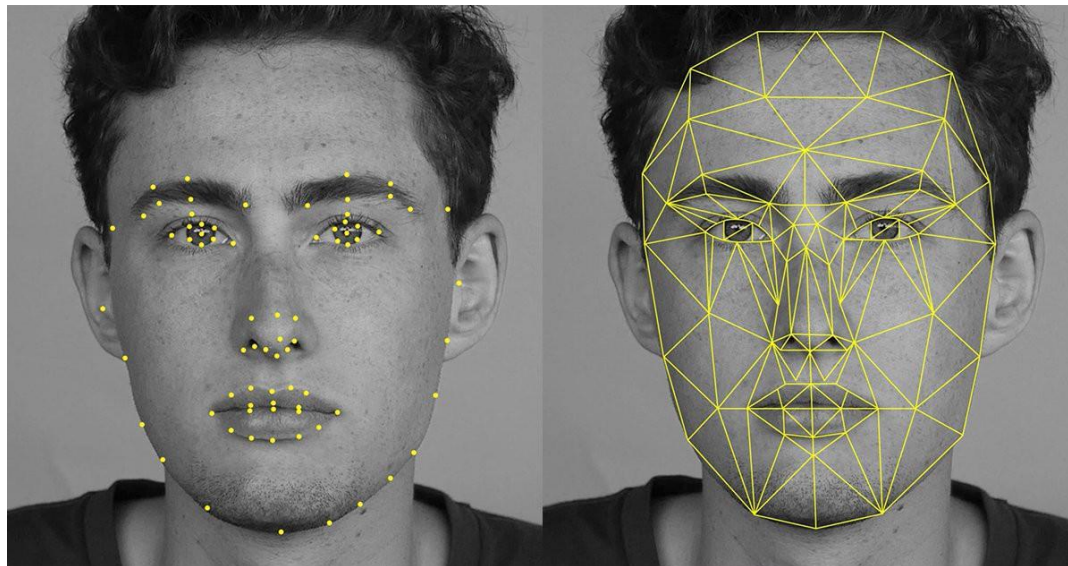
Edge Applications

- Shopping with Augmented Reality
- Smart-X Services
- Image Analytics





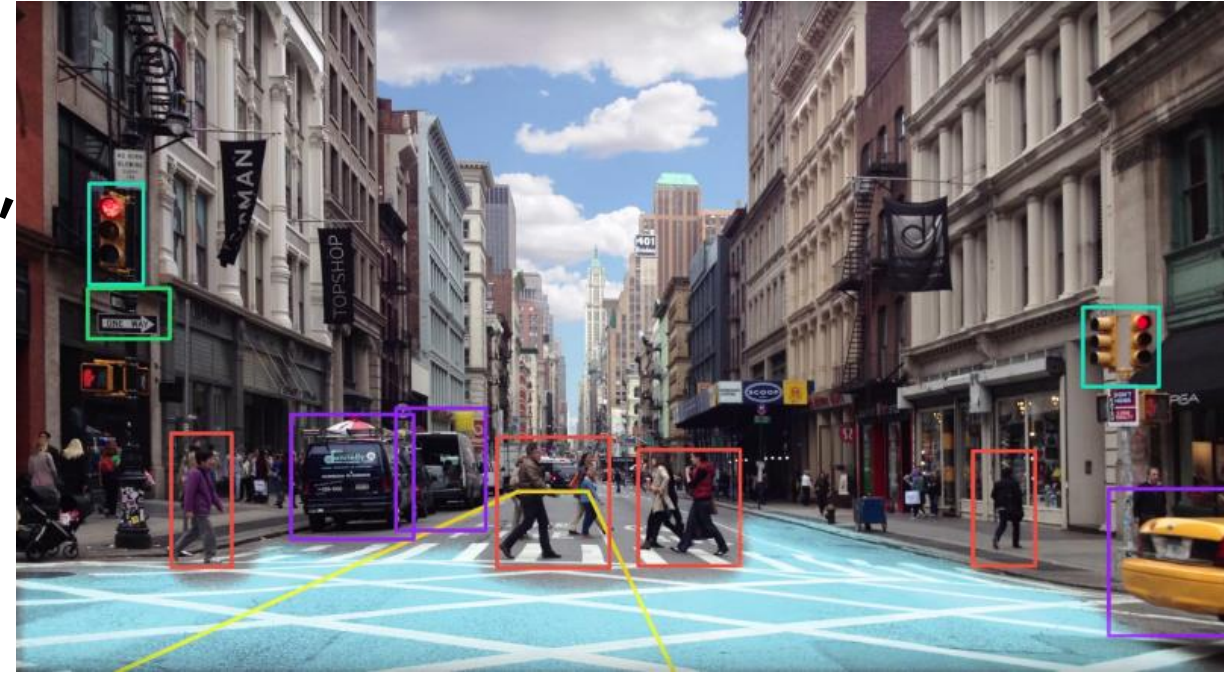
- Real-time recognition of a constantly changing scene based on video streaming requires high data bandwidth if performed in the cloud.
- AI on the Edge enables local analysis of the visual scene in various flavors, such as
 - Understanding the scene for context analysis,
 - Simultaneous multi-object detection and
 - Recognition for obstacle avoidance,
 - People identification for secure access, and more.



- Surveillance and Monitoring: Deep Learning-enabled smart cameras could locally process captured images to identify and track multiple objects and people, detecting suspicious activities directly on the edge node.
- Smart cameras minimize communication with the remote servers by only sending data on a triggering event, also reducing remote processing and memory requirements.
- Intruder monitoring for secure homes and monitoring of elderly people are typical applications.



- **Autonomous Vehicles:** A smart automotive camera can recognize vehicles, traffic signs, pedestrian, road, and objects locally, sending only information needed to perform autonomous driving to the main controller.
- A similar concept can be applied to robots and drones.

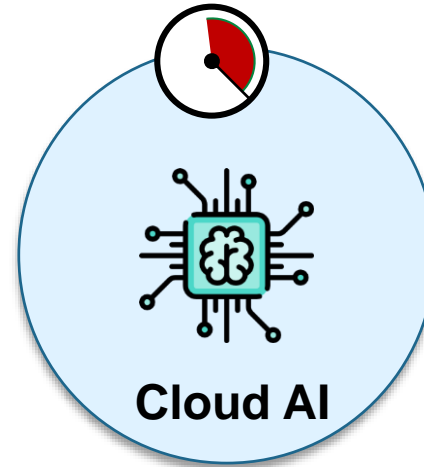


Enabling Technologies to Implement Edge AI

- **Cloud VS. Edge AI**
- **Machine Learning Taxonomy - Type of Machine Learning Schemes for Edge AI**
- **Supervised learning**
- **Unsupervised Learning**
- **Reinforcement Learning**

Pros

- Can train large neural network model
- High computation resources
- Big Data processing
- Easy to scale
- Low cost storage

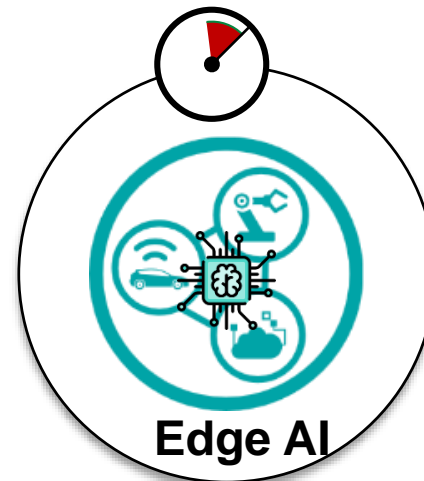


Cons

- High service delay
- High bandwidth cost
- Sending raw data over the Internet to the cloud have privacy, security and legal issues

Pros

- Real-time predictions for mission-critical applications
- Efficient use of network bandwidth
- Process data closest to the source
- Low latency response
- Support mobility



Cons

- Low computation power than the cloud
- Computation resources are Less scalable than cloud

Schemes for Edge AI

Machine Learning Types

Supervised learning

SVM, kNN, Naïve Bayes,
Random Forest

Continuous target variable

Regression

Logistic
Linear

Housing price prediction

Categorical target variable

Classification

SVM
kNN

Medical Imaging

Unsupervised learning

K-means, HMM, CRF,
MEMM, GMM

No target variable

Clustering

K-means
Hierarchical

Customer Segmentation

Association

Apriori
DBSCAN

Market Basket Analysis

Reinforcement learning

MDP, Markov approximation, Q-learning

Categorical target variable

Classification

Hybrid
RL+SVM
RL+NN

Optimized Marketing

No target variable

Control

Deep Q-learning
Actor Critic learning

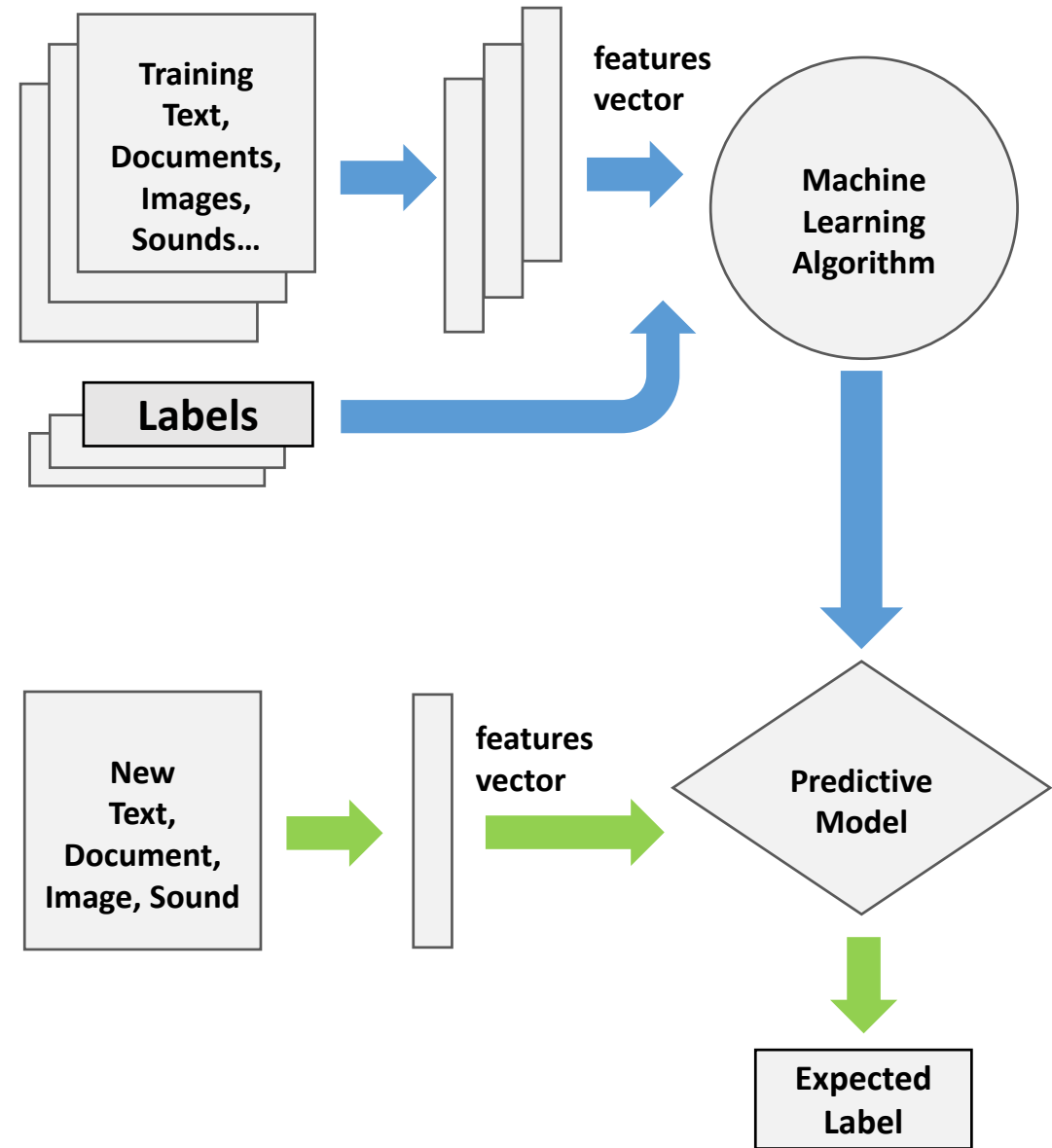
Self-Driving Cars

Supervised Learning - relies on data where the true label is indicated.

- Example: teaching a computer to distinguish between pictures of cats and dogs, with each image tagged “cat” or “dog”.
- Labeling is normally performed by humans to guarantee high data quality.
- Having learned the difference, the ML algorithm can now classify new data and predict labels (“cat” or “dog”) on previously unseen images.

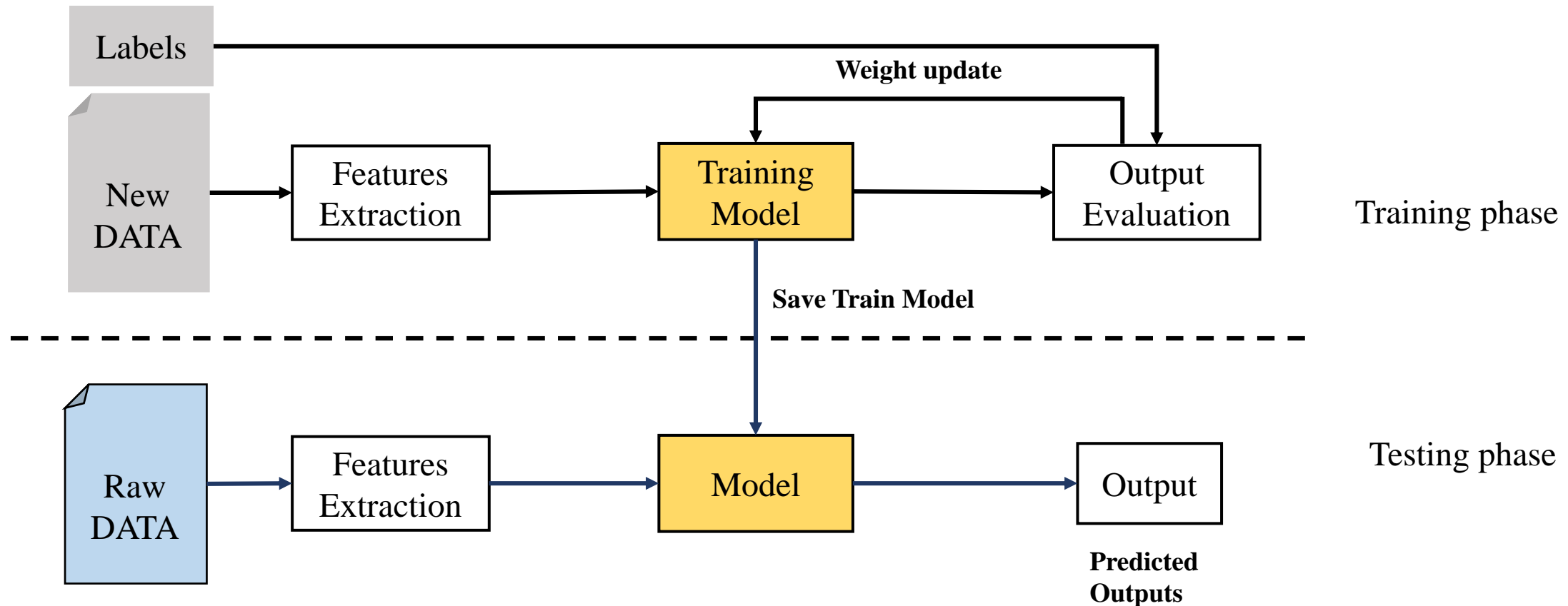


- Supervised learning algorithms are trained using labeled data.
- When dealing with labeled data, both the input data and its desired output data are known to the system.
- Supervised learning is commonly used in applications that have **enough historical data**.
- Applications:
 - Classification
 - Regression



Feature Extraction : Scaling (normalized inputs)
Dimensionality Reduction
Feature Selection : Selecting important features

- Model Selection – Convolutional Neural Network, Recurrent Neural Network, etc..
- Performance Metrics – Accuracy
- Hyper parameter Optimization – Weight, Bias, Hidden layers, etc..
- Cross-Validation - Cross-validation is a technique to evaluate predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it.



- Algorithms:
 - Decision tree
 - Random forest
 - Neural networks
 - Support vector machines
 - Ensemble learning
 - Bayesian learning
 - And so on..
- Examples:
 - Speech recognition used in smart devices is based on supervised learning techniques.

- Supervised Learning is being used to detect spam emails, i.e., Naive Bayes spam filtering. Particular words have specific probabilities of occurring in spam email and in legitimate email, e.g., “refinance”, “Viagra”
- Probabilities are not known in advance. A filter is trained by users manually indicating if email is spam or not through which the filter adjusts the probabilities of each word and save in its database
- After training, the word probabilities are used to compute the probability that an email with a particular set of words belongs to either spam or not spam category

Input

$x \in \mathcal{X}$

Output

$y \in \mathcal{Y}$

Spam
filtering

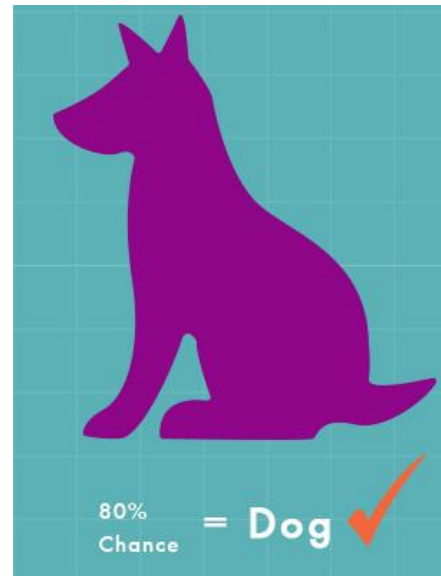


Spam
or
Not Spam

Unsupervised Learning

- Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data.
- No labels are given to the learning algorithm, leaving it on its own to find structure in its input.
- Unsupervised learning can be a goal in itself **to discover hidden patterns in data.**

Example: presented with images of cats and dogs that have not been labeled, unsupervised ML can separate the images into two groups based on some inherent characteristics of the images.

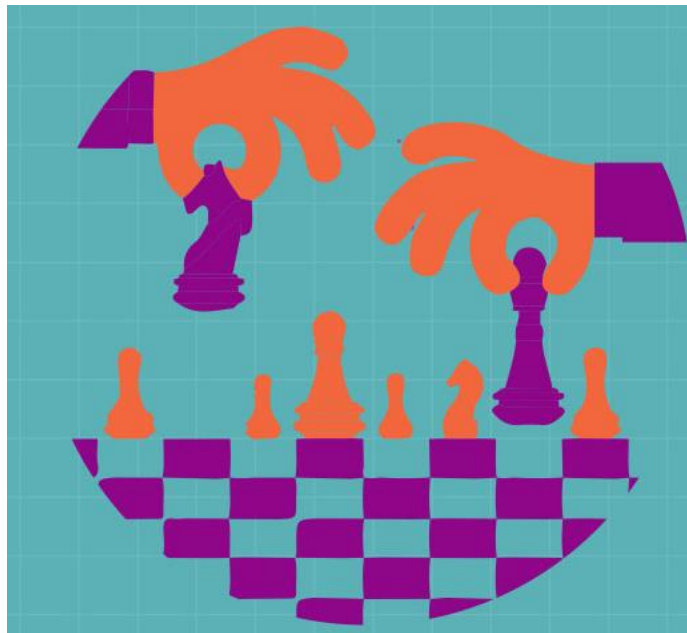


Applications:
Clustering
Associations
Anomaly detection

- Algorithms:
 - Kmeans clustering
 - Hierarchical clustering
 - DBScans
 - Apriori Associations
 - Principal component analysis
 - Independent component analysis
 - Non-negative matrix factorization
 - And so on..
- Examples:
 - Market segmentation uses clustering to identify subgroups of people who might be more receptive to a specific form of advertising, or more likely to purchase a particular product.
 - In medicine, clustering diseases, cures for diseases, or symptoms of diseases can lead to very useful taxonomies.

- Reinforcement learning:
 - Reinforcement learning is a type of learning in which an agent learns its best action **through trial-and-error by interactions with a dynamic environment.**
 - Reinforcement Learning is learning how to act in order to maximize a numerical reward.
 - Close to human learning.
 - Every action has some impact in the environment, and the environment provides rewards that guides the learning algorithm.
- Applications:
 - Delivery Management
 - Supply chain inventory management
 - Stock market trading
 - Mobile network applications

- Reinforcement Learning – Example: learning to play chess. ML receives information about whether a game played was won or lost. The program does not have every move in the game tagged as successful or not, but only knows the result of the whole game. The ML algorithm can then play a number of games, each time giving importance to those moves that result in a winning combination.

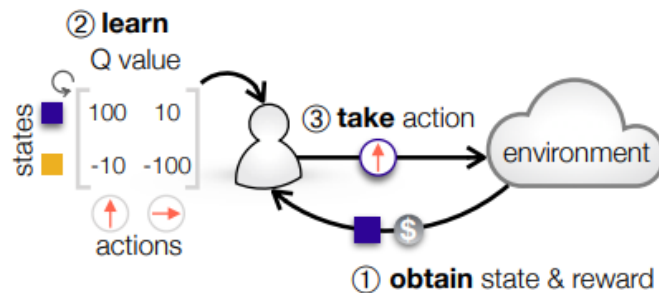


- Algorithms:
 - Q-Learning
 - Double Q-Learning
 - Actor critic learning
 - State–action–reward–state–action (SARSA)
 - Expected SARSA
 - Temporal-Difference Learning
 - And so on..

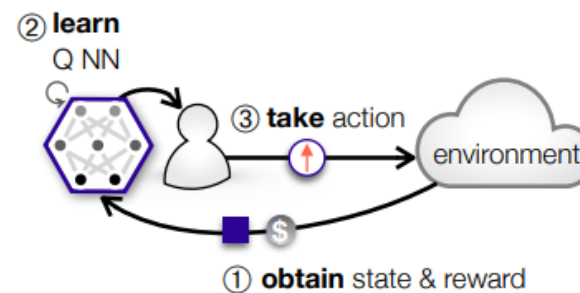
- Examples:
 - A robot uses reinforcement learning to pick a device from a box and put it in a container. During learning process, it succeeds and fails, however, it memorizes the rewards and train's itself to do this job with great speed and precision after a certain time. Such robots can be used in house cleaning and management.

The goal of RL is make an agent in an environment take an optimal action at a given current state, where the interaction between the agent's action and state through the environment is modeled as a Markov decision process (MDP). [9]

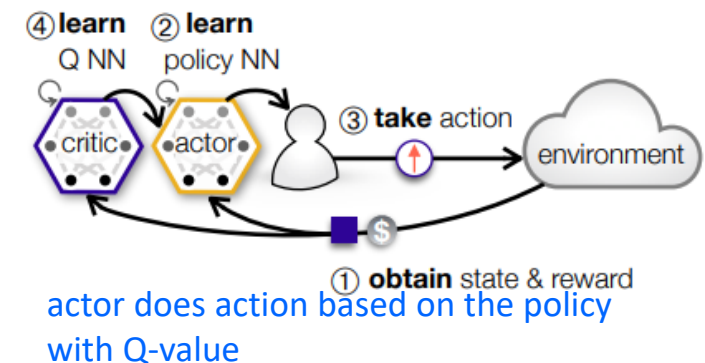
- When each action is associated with a return, the agent takes an action that maximizes its predicted cumulative return, e.g., Q-learning that maximizes the Q value for each state, as illustrated in Figure(a).
- In Q-learning, the larger state dimension, the more computation. This problem is resolved by deep Q-learning as shown in Figure (b), where a NN approximates the Q function and produces the Q values by feeding a state. These value-based RL can take actions only through Q values that are not necessarily required. Instead, one can directly learn a policy that maps each state into the optimal action, which is known as policy-based RL whose variance may become too large [10].
- Actor-critic RL is a viable solution to both problems, comprising a Neural Network (NN) that trains a policy (actor NN) and another NN that evaluates the corresponding Q value (critic NN), as visualized in Figure (c).



(a) Classical Q-learning.



(b) Deep Q-learning.



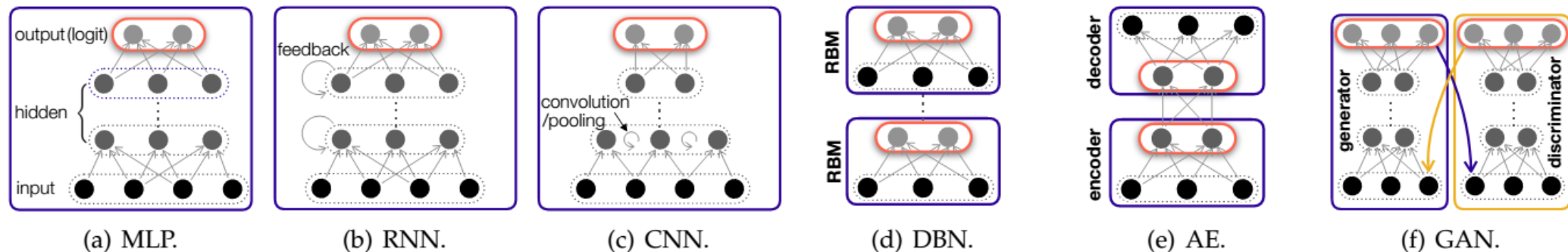
(c) Actor-critic RL.

Examples of RL: (a) classical Q-learning without any Neural Network; (b) deep Q-learning with a Neural Network, and (c) actor – critic RL with actor and critic Neural Networks.

[9] Park, J., Samarakoon, S., Bennis, M. and Debbah, M., 2018. Wireless network intelligence at the edge. *arXiv preprint arXiv:1812.02858*.

[10] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in Proc. of the 12th NIPS, NIPS'99, (Colorado, USA), pp. 1057–1063, MIT Press, Dec. 1999

- It requires high computation resources to process the big data (high-dimensional data) to train the prediction models.
- It is difficult to find a suitable prediction model among the various types of deep learning models, such as Multilayer Perceptron (MLPs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Convolutional Recurrent Neural Networks (CRNNs), etc. [11].
- It is difficult to tune parameters such as the number of layers (i.e., the depth of the network), types of layers (e.g., Convolutional, Recurrent and Fully Connected layers), and learning rate to improve the accuracy of the prediction model.



Types of NN architectures: (a) multilayer Perceptron (MLP); (b) recurrent neural network (RNN); (c) convolutional neural network (CNN); (d) deep belief network (DBN) with restricted Boltzmann machines (RBMs); (e) auto encoder; and (f) generative adversarial network (GAN) [3]

[9] Park, J., Samarakoon, S., Bennis, M. and Debbah, M., 2018. Wireless network intelligence at the edge. *arXiv preprint arXiv:1812.02858*.

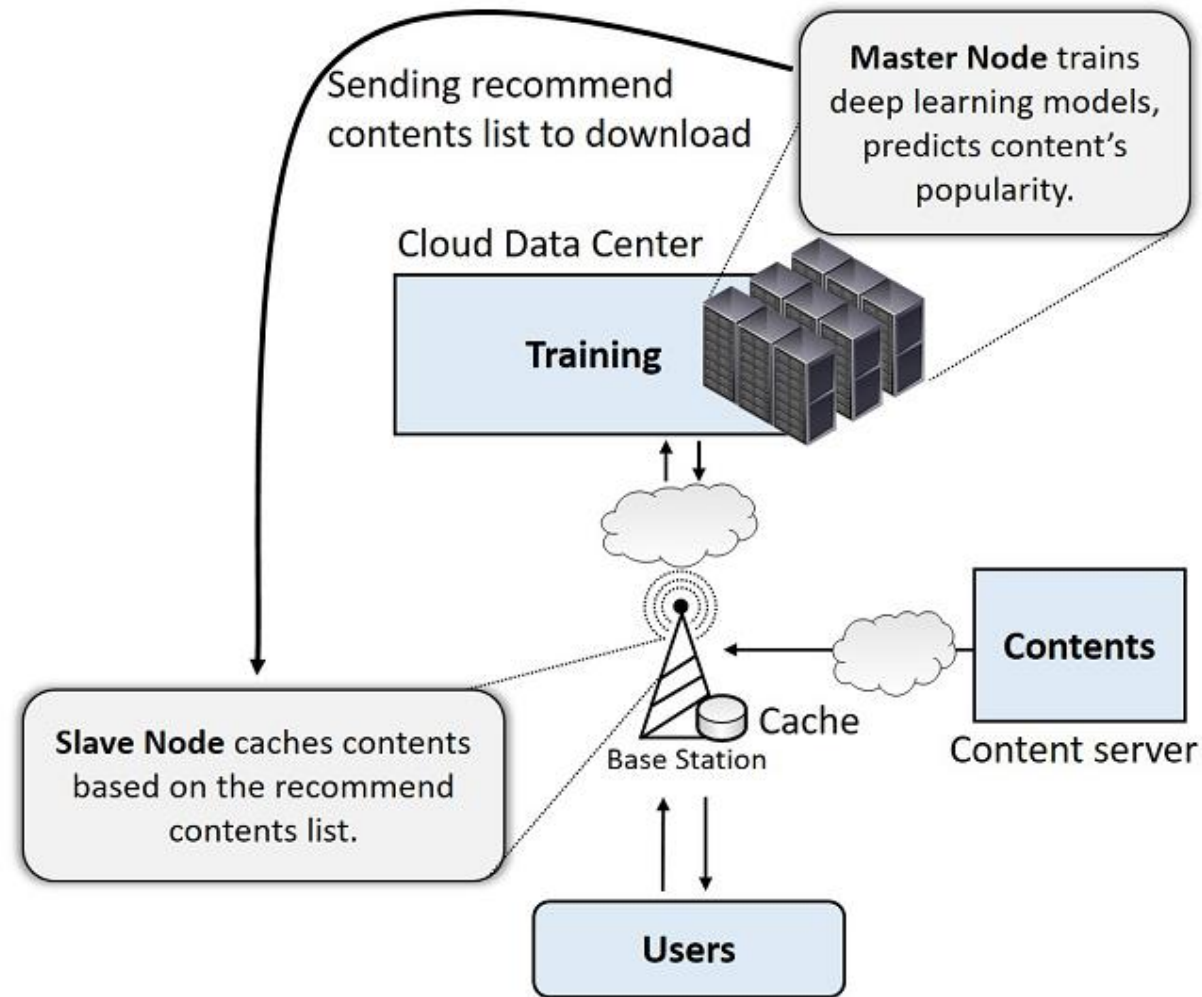
[11] I. Goodfellow, Y. Bengio and A. Courville, "Deep Learning", in MIT Press, 2016.

Use Cases

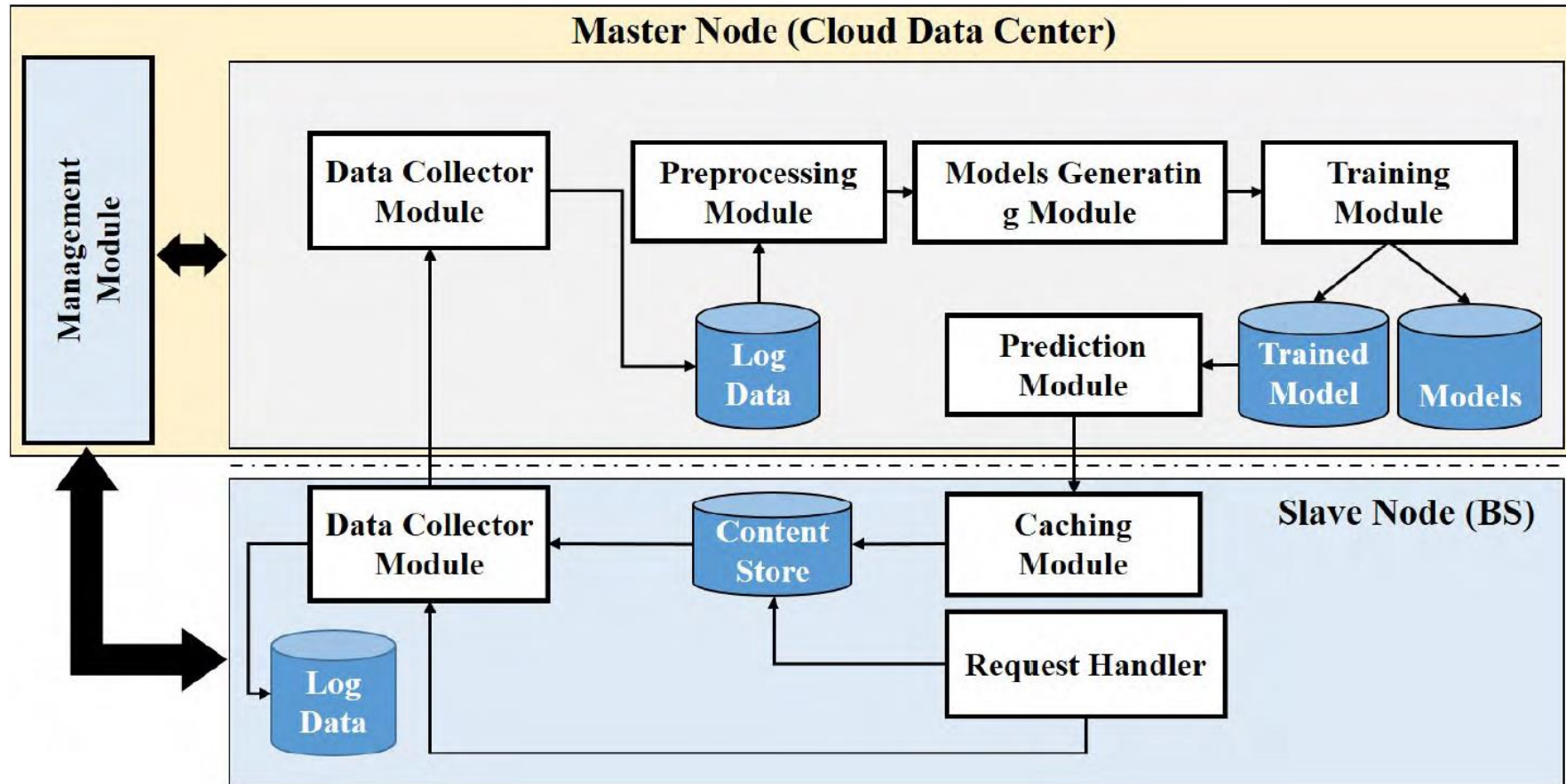
- **Use Case-1 : Content Caching at Edge**
- **Use Case-2: Content Caching at Virtualized Edge**
- **Use Case-3: License Plate Detection**

- Caching popular contents at edge nodes such as base stations is a crucial solution for improving users' quality of services in next generation networks.
- However, it is very challenging to correctly predict the future popularity of contents and decide which contents should be stored in the base station cache.
- Recently, with the advances in big data and high computing power, deep learning models have achieved high prediction accuracy.
- **Goal** : Maximize the cache hit, in order to reduce access latency
- **Potential Benefits** : Enhanced cache hit, low access latency, bandwidth saving for backhaul
- **Approach**: Randomized deep learning model search to get popularity prediction model
 - **Input** : Features information such as number of request
 - **Output** : Content's popularity score
 - **Dataset** : MovieLens (<https://grouplens.org/datasets/movielens/>)

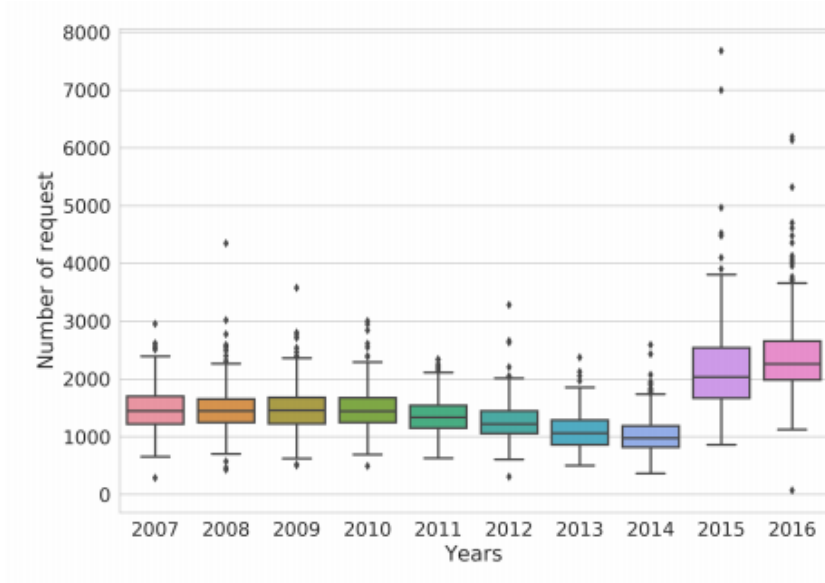
- System Model



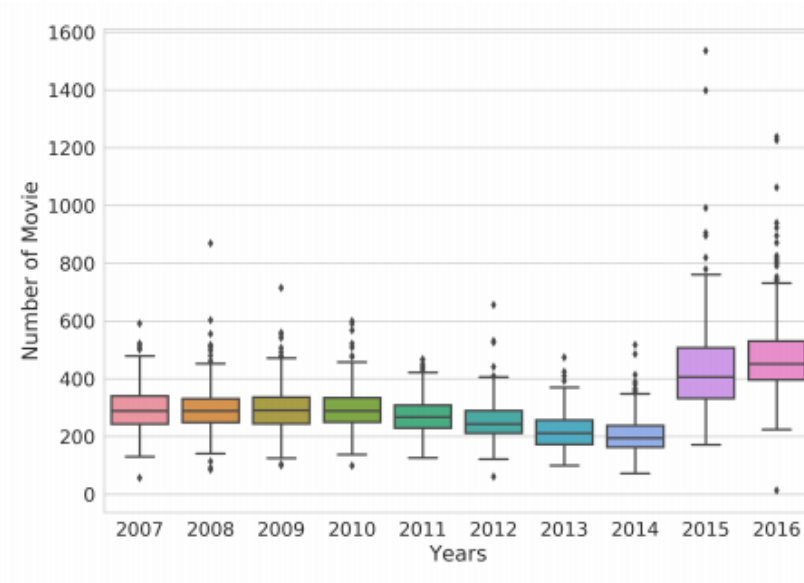
System model of learning-based caching at the edge.



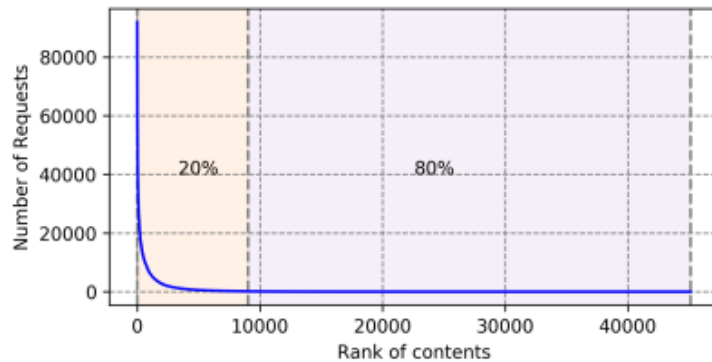
An overview system design for learning-based caching at the edge.



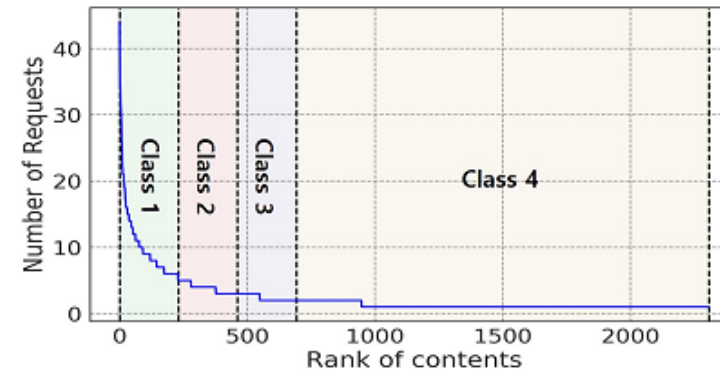
Daily request count of all contents (movies) for each year.



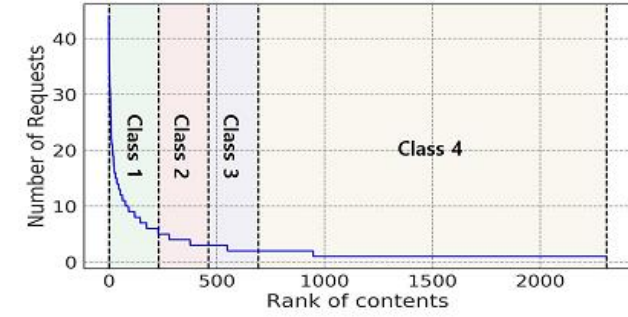
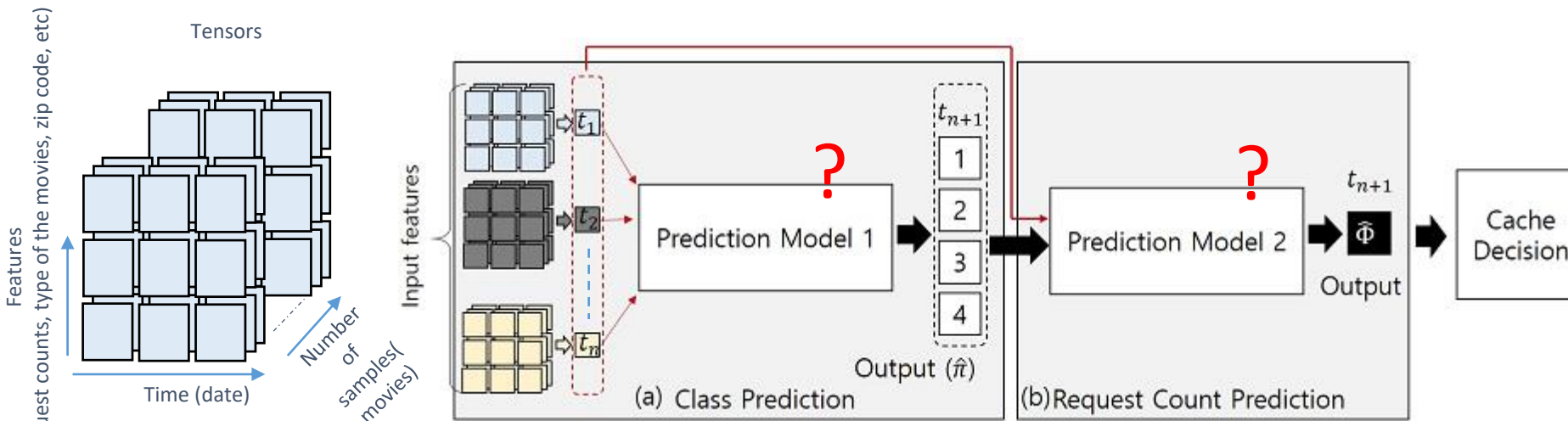
Daily request count of top 20% contents (movies) for each year.



Ranking of all contents (movies) based on total request count received.

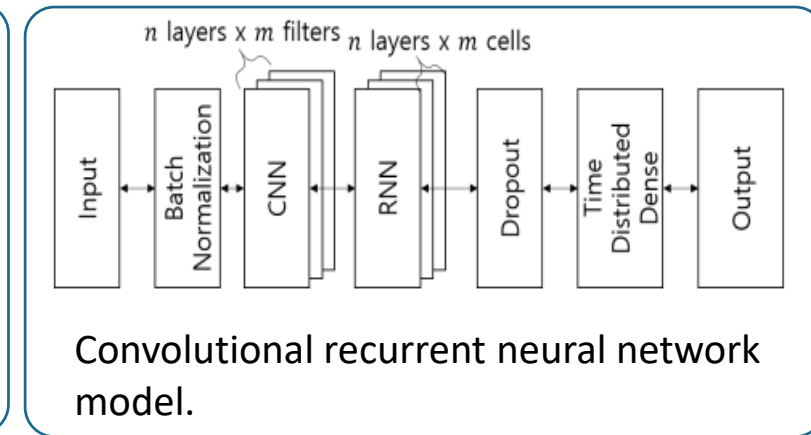
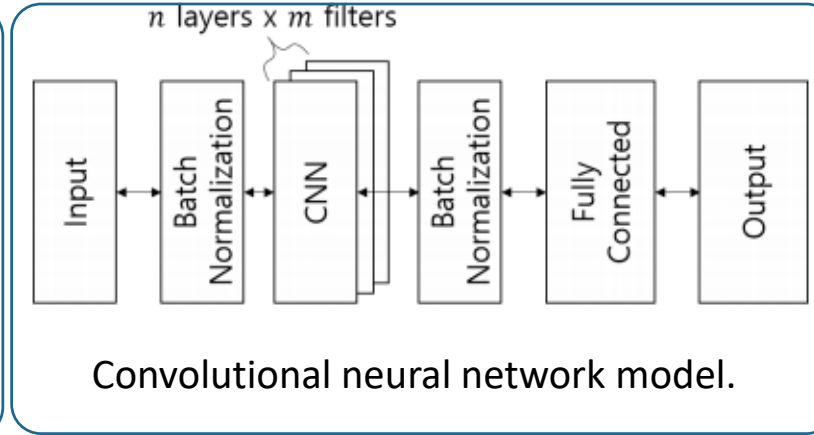
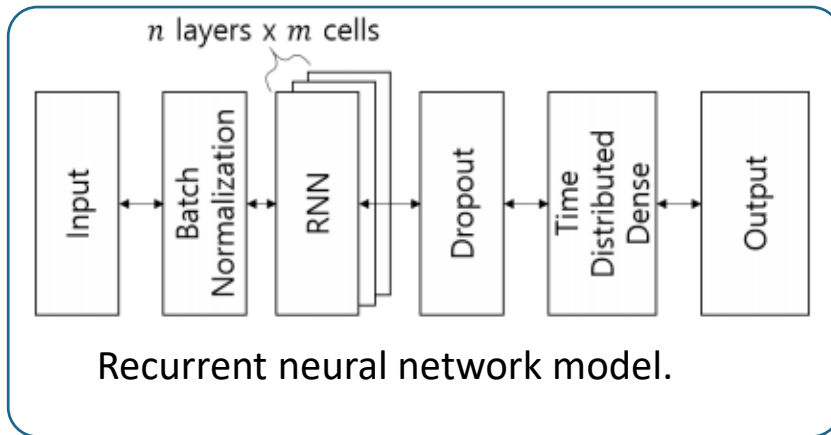


Ranking of all contents (movies) based on daily received content requests and classification labels.

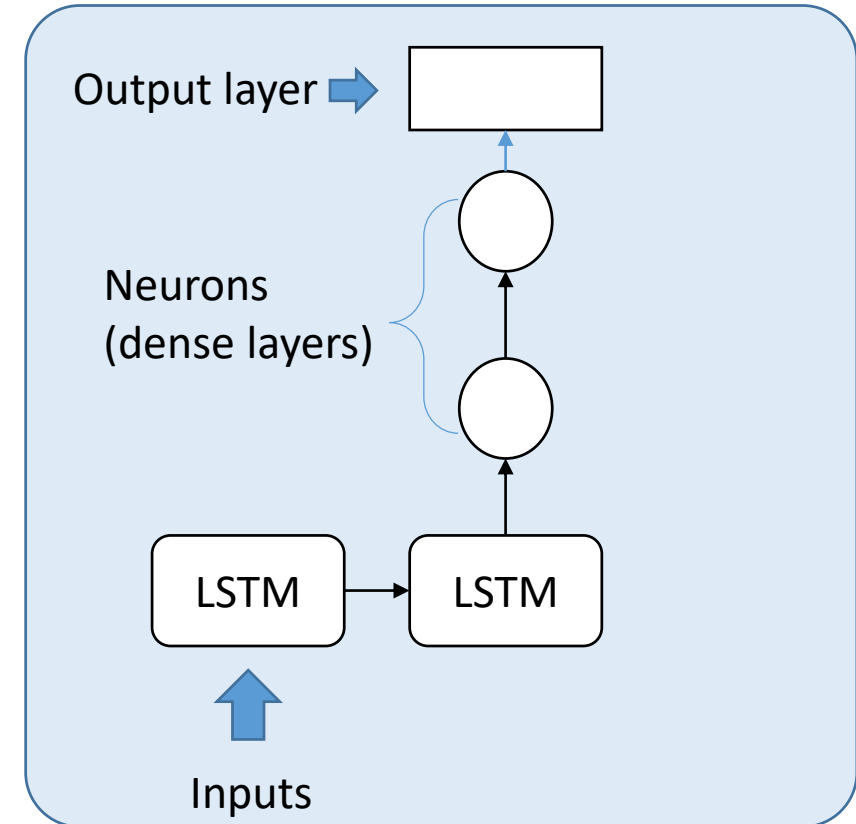
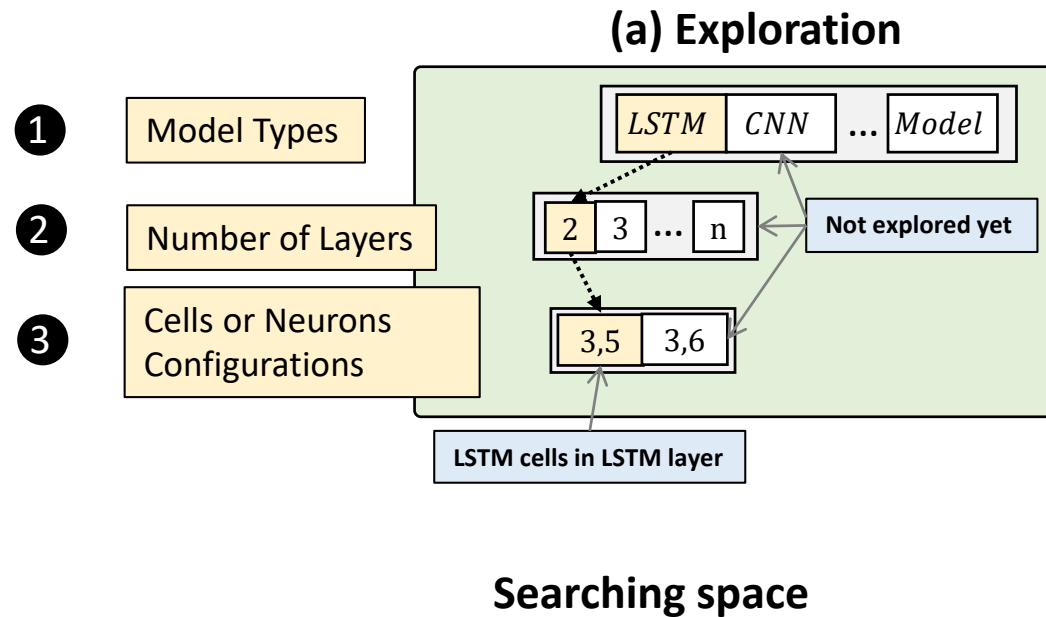


The overview prediction model used in the proposed scheme: (a) Class prediction, (b) Request count prediction (for the specific movie).

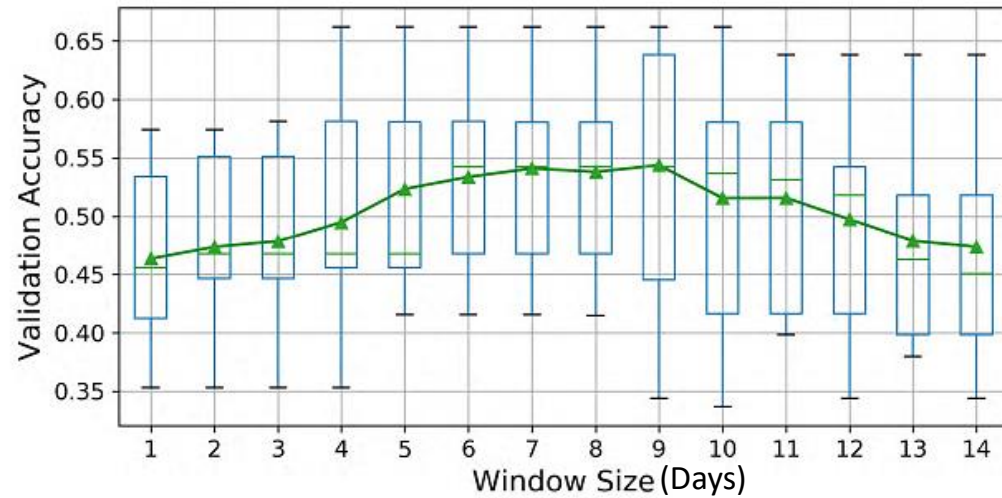
Randomized Model Search



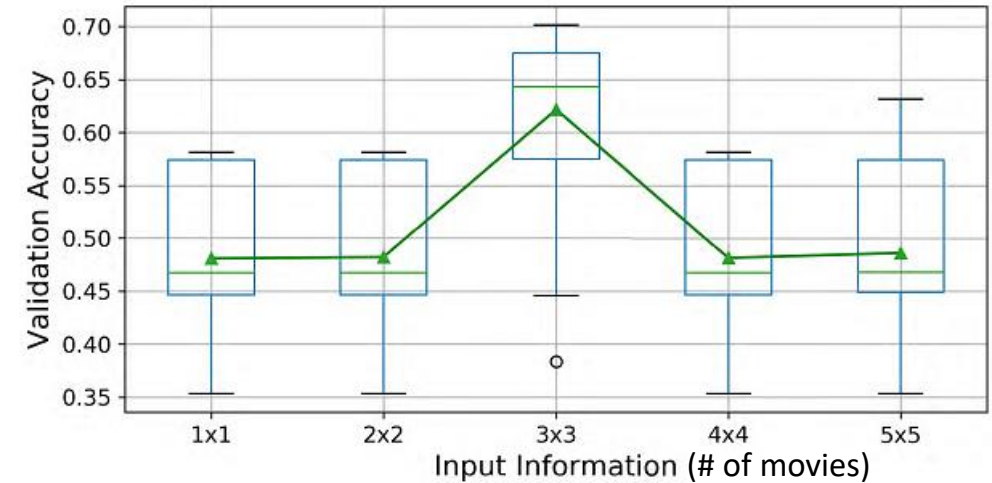
Randomized Model Searching Approach



- Random-search finds the best-suited model among random configurations in order to reduce searching space.
- But, random-search method explores in a random direction to find the best deep learning model within possible configurations.



Finding a suitable number of sequences data.



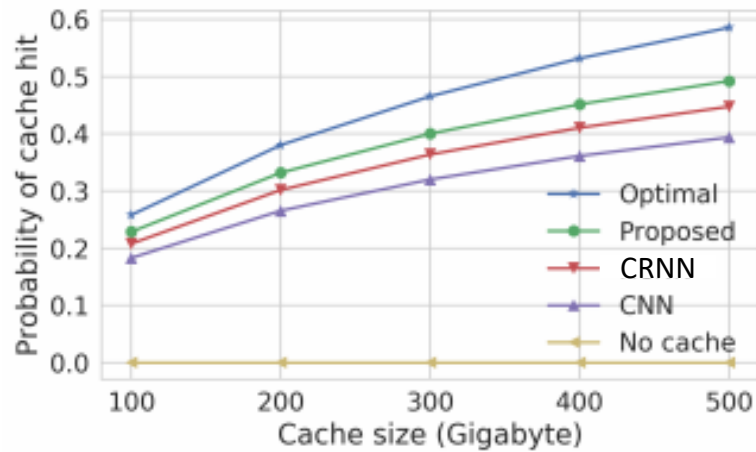
Prediction accuracy based on the neighbor's information.

Model Type	Configuration	Validation accuracy	Computing Time (s)
CRNN (Class)	Conv[2,4], LSTM[4]	0.643	7.258
LSTM (Class)	LSTM[4,1]	0.747	1.302
CNN (Class)	Conv[4,4,2], FC[1]	0.724	2.354
Model Type	Configuration	Validation loss	Computing Time (s)
CRNN (Req)	Conv[3], LSTM[3,5]	0.059	8.853
LSTM (Req)	LSTM[3,5]	0.056	17.547
CNN (Req)	Conv[1,4,3], FC[3]	0.066	22.160

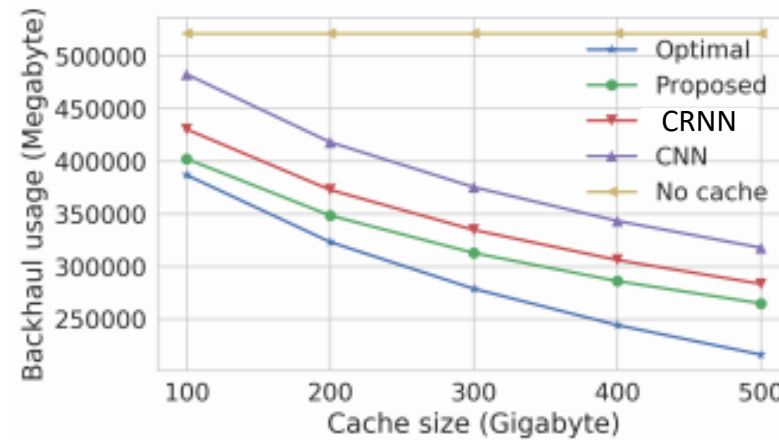
Cache demand

Content's Popularity

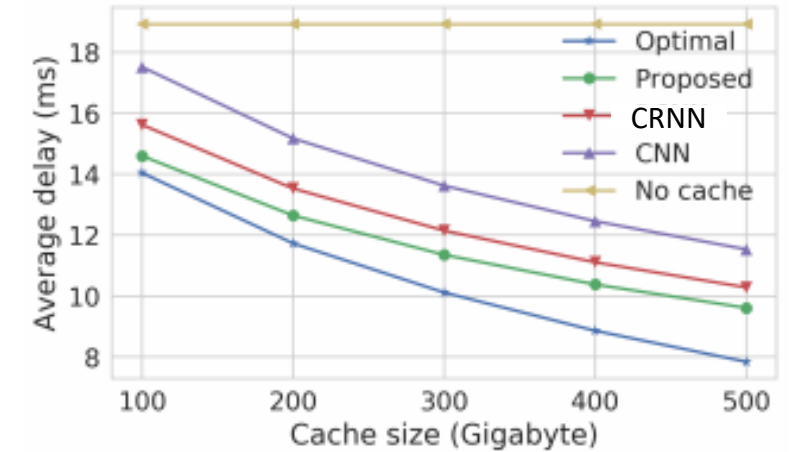
Neighbor = similar contents based on Zipf ranking



(a)



(b)

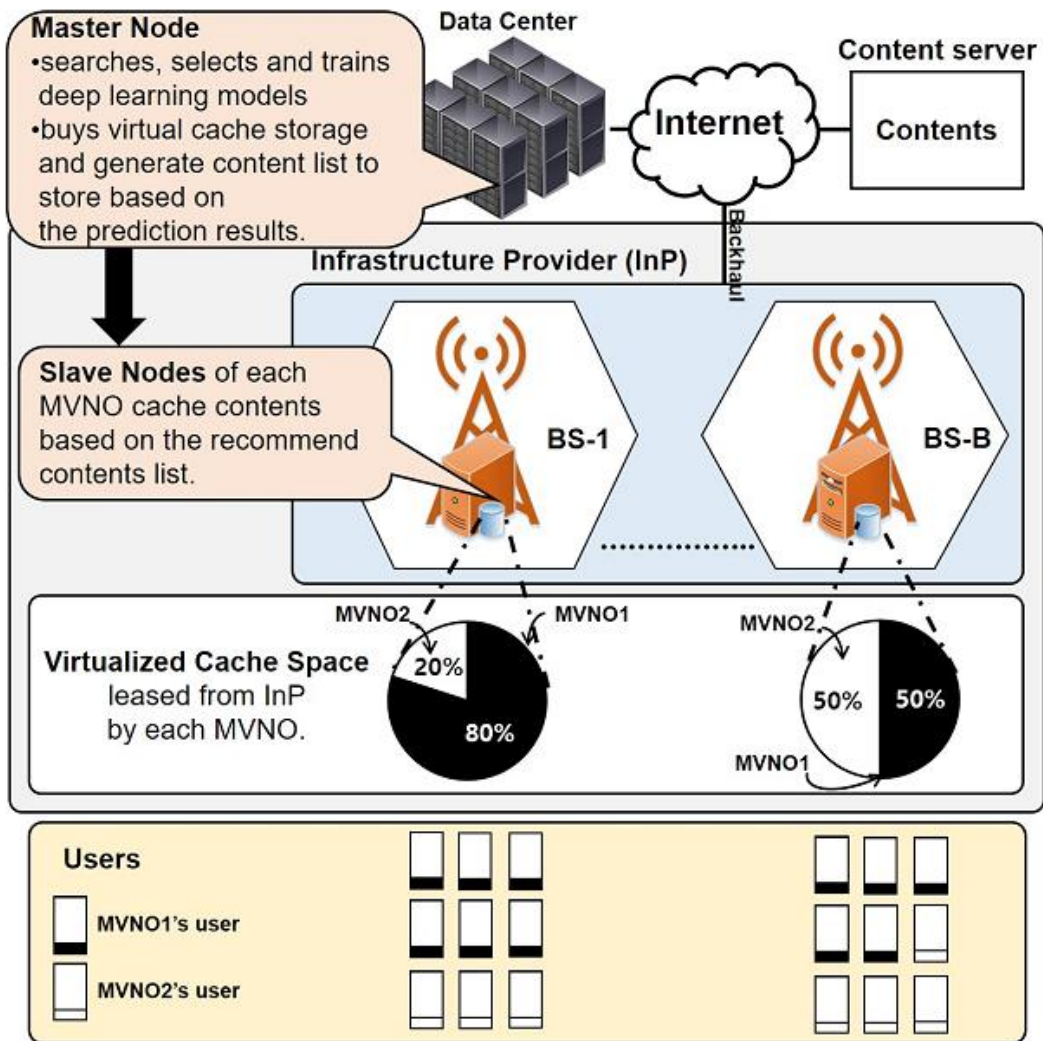


(c)

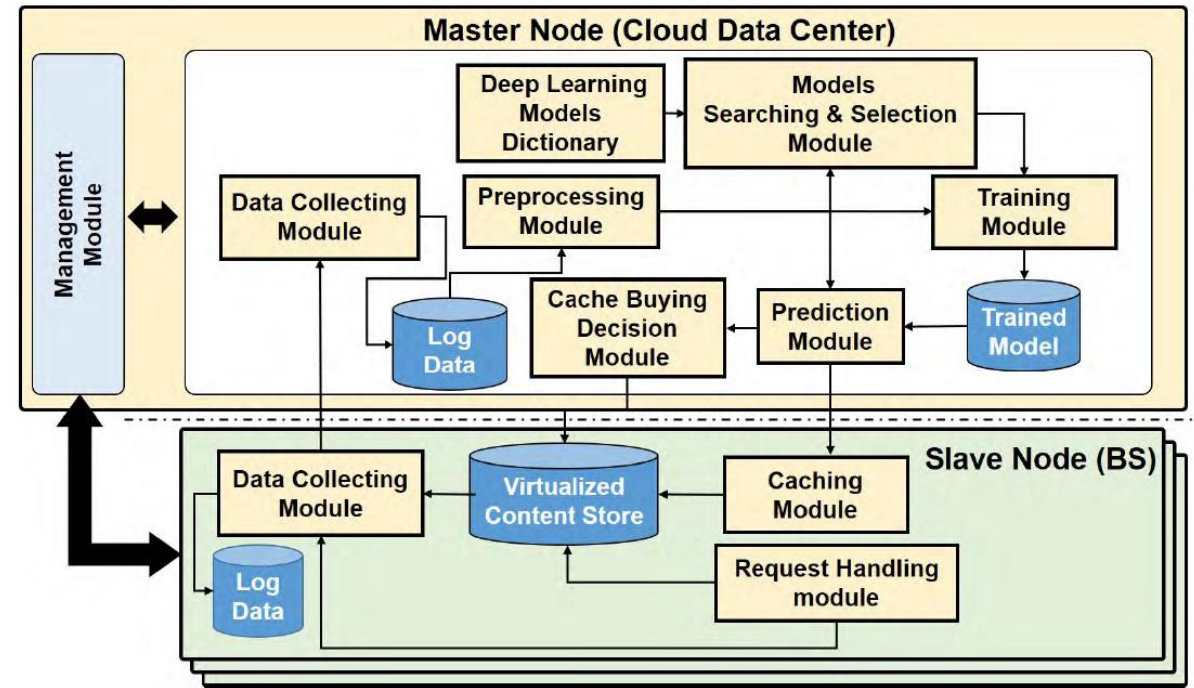
Caching related performance comparisons: (a) Cache hit, (b) Backhaul usage comparison, and (c) Access delay comparison.

- Goal : Maximize the cache hit, in order to reduce access latency/ To improve the profit of MVNO
- Potential Benefits : Enhanced cache hit, low access latency, bandwidth saving for backhaul
- Approach: Reinforcement learning based deep learning model search to get the “cache demand prediction model” and content’s popularity prediction model.
 - Input : Features information such as number of request
 - Output : Cache demand and Content’s popularity score

MVNO : Mobile Virtualized Network Operator

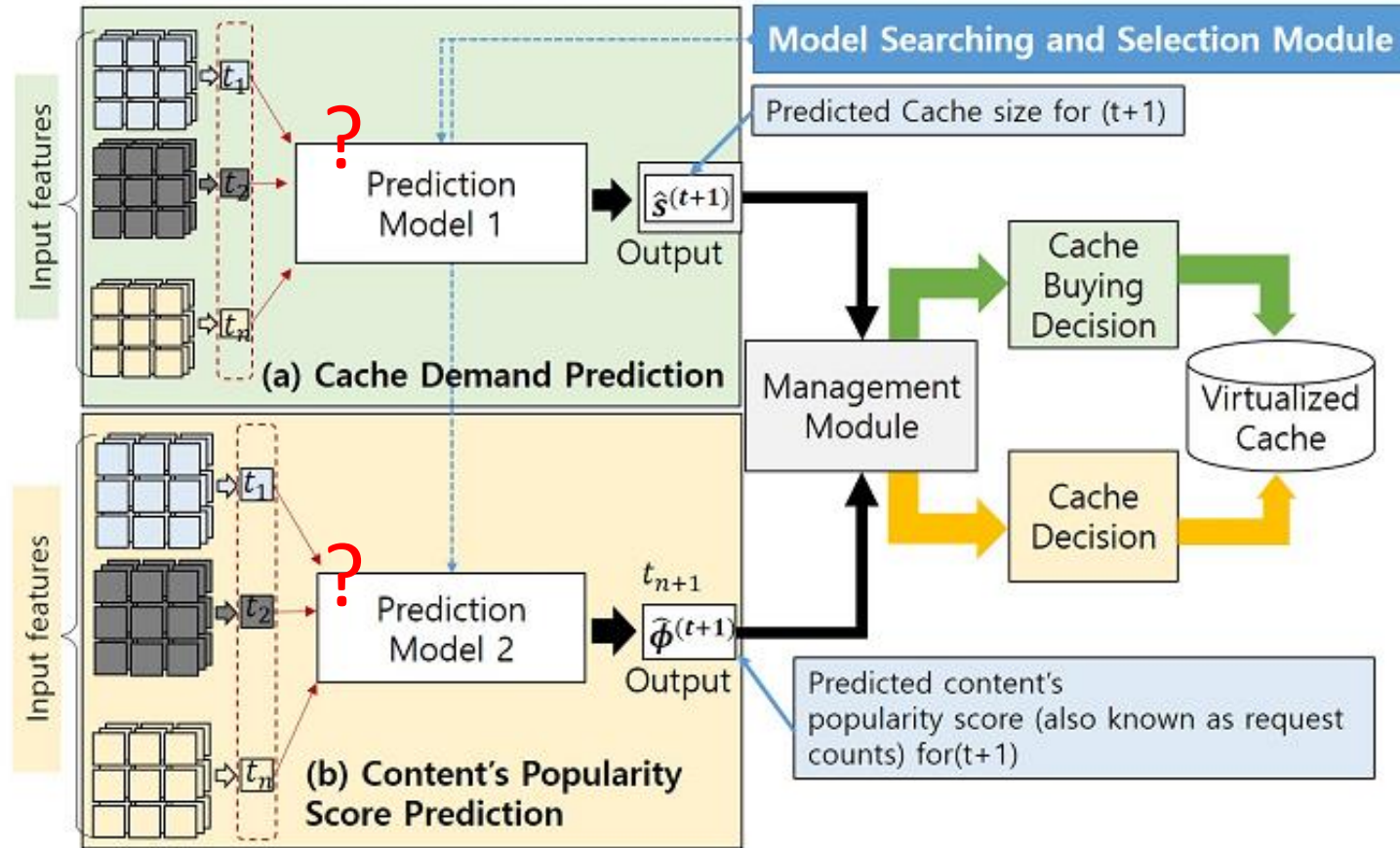


Virtualized cache management for MVNO.

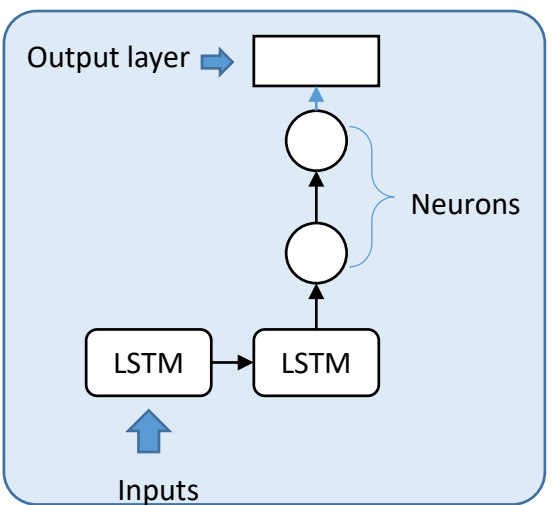
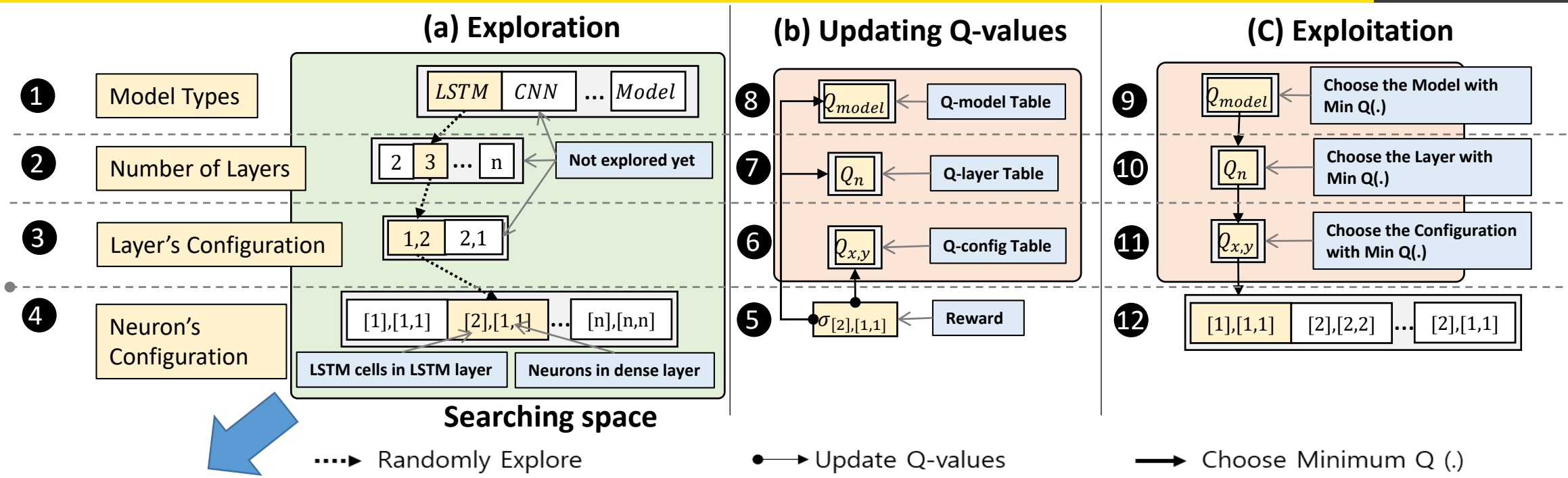


An overview system design for learning-based caching at the edge.

[13] Kyi Thar, Thant Zin Oo, Yan Kyaw Tun, Do Hyeon Kim, Ki Tae Kim, and Choong Seon Hong, "A Deep Learning Model Generation Framework for Virtualized Multi-access Edge Cache Management,"



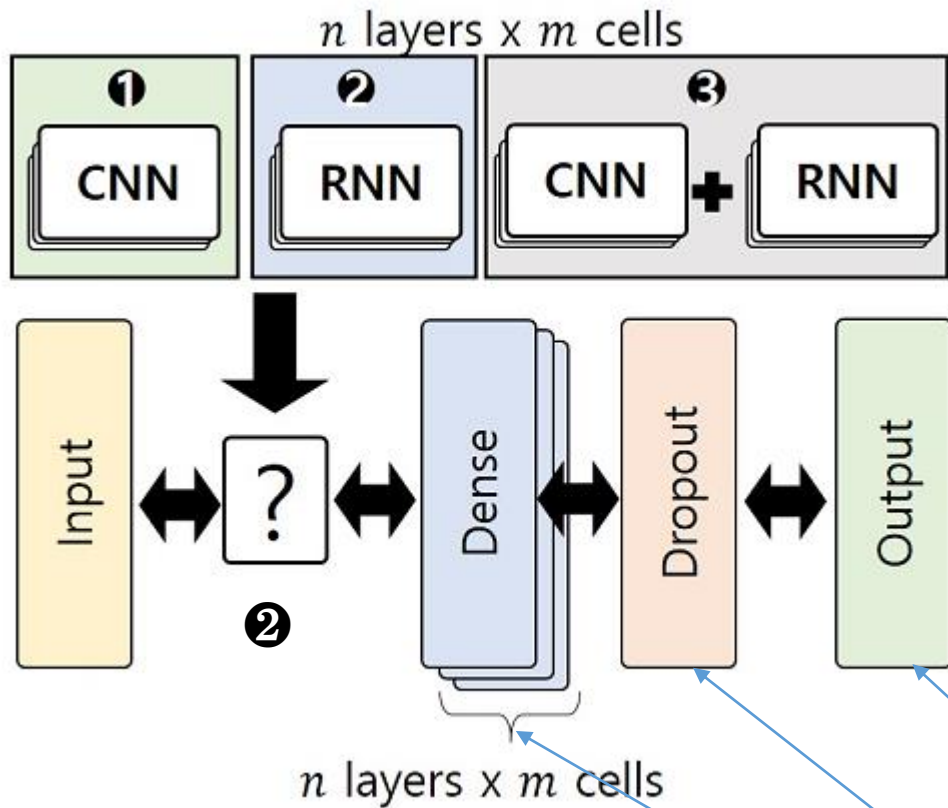
The overview prediction model used in the proposed scheme: (a) Cache demand prediction, (b) Content's popularity scores prediction.



Reward = Training Loss

Choose Max Q(.) when the objective is to improve the accuracy

Choose Min Q(.) when the objective is to reduce the training loss

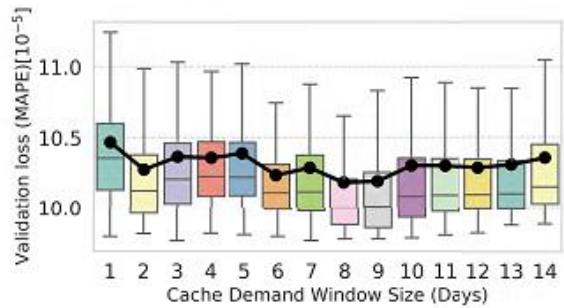


Deep Learning Models Framework.

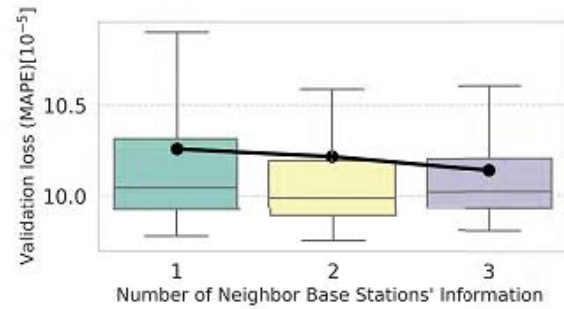
Summary of output layer activation function.

Activation	Equation	Range
Softmax	$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$	[0,1]
Sigmoid	$\text{Sigmoid}(z_i) = \frac{1}{1+e^{-z}}$	[0,1]
Tanh	$\text{Tanh}(z_i) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	[-1,1]
Relu	$\text{Relu}(z_i) = \begin{cases} 0, & \text{for } z_i < 0. \\ z, & \text{for } z_i \geq 0. \end{cases}$	[0,1]

- Output layer is configured depending on the prediction problem
- To prevent over-fitting problem
- To extract the features information

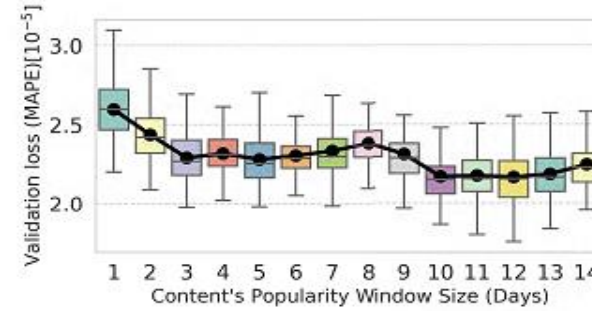


Finding the best window size for **cache demand prediction**

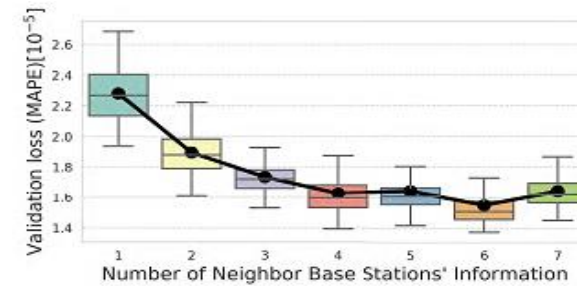


Finding the best number of neighbors (Base Station)

(b)



Finding the best window size for content's **popularity prediction**



Finding the best number of neighbors (Base Station)

(b)

Model	Model Type	Configuration	Validation loss	Computing Time (s)
M 1	LSTM (Cache)	LSTM ⟨60⟩, Dense ⟨283⟩	0.0000995	0.194
M 2	LSTM (Cache)	LSTM ⟨82⟩, Dense ⟨165, 217⟩	0.0000998	0.231
M 3	CNN (Cache)	CNN ⟨48, 170⟩, Dense ⟨52⟩	0.724	0.00001
M 1	LSTM (Req)	LSTM ⟨73⟩, Dense ⟨176, 106, 64, 143⟩	0.000016	0.333
M 2	LSTM (Req)	LSTM ⟨53⟩, Dense ⟨63, 64, 67, 127⟩	0.000017	0.272
M 3	LSTM (Req)	LSTM ⟨78⟩, Dense ⟨157, 27, 112, 112, 169⟩	0.000018	0.346

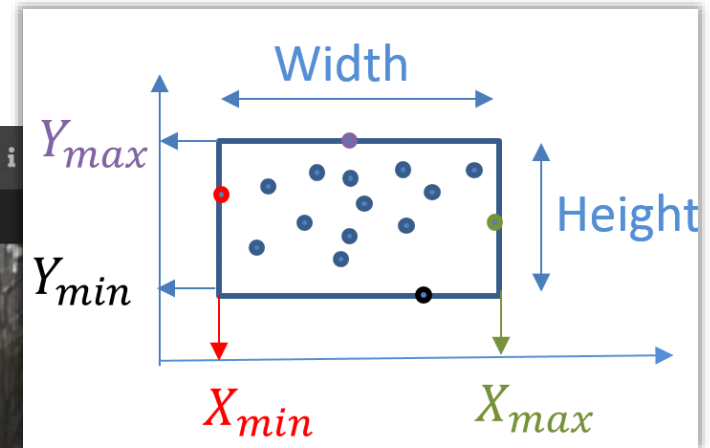
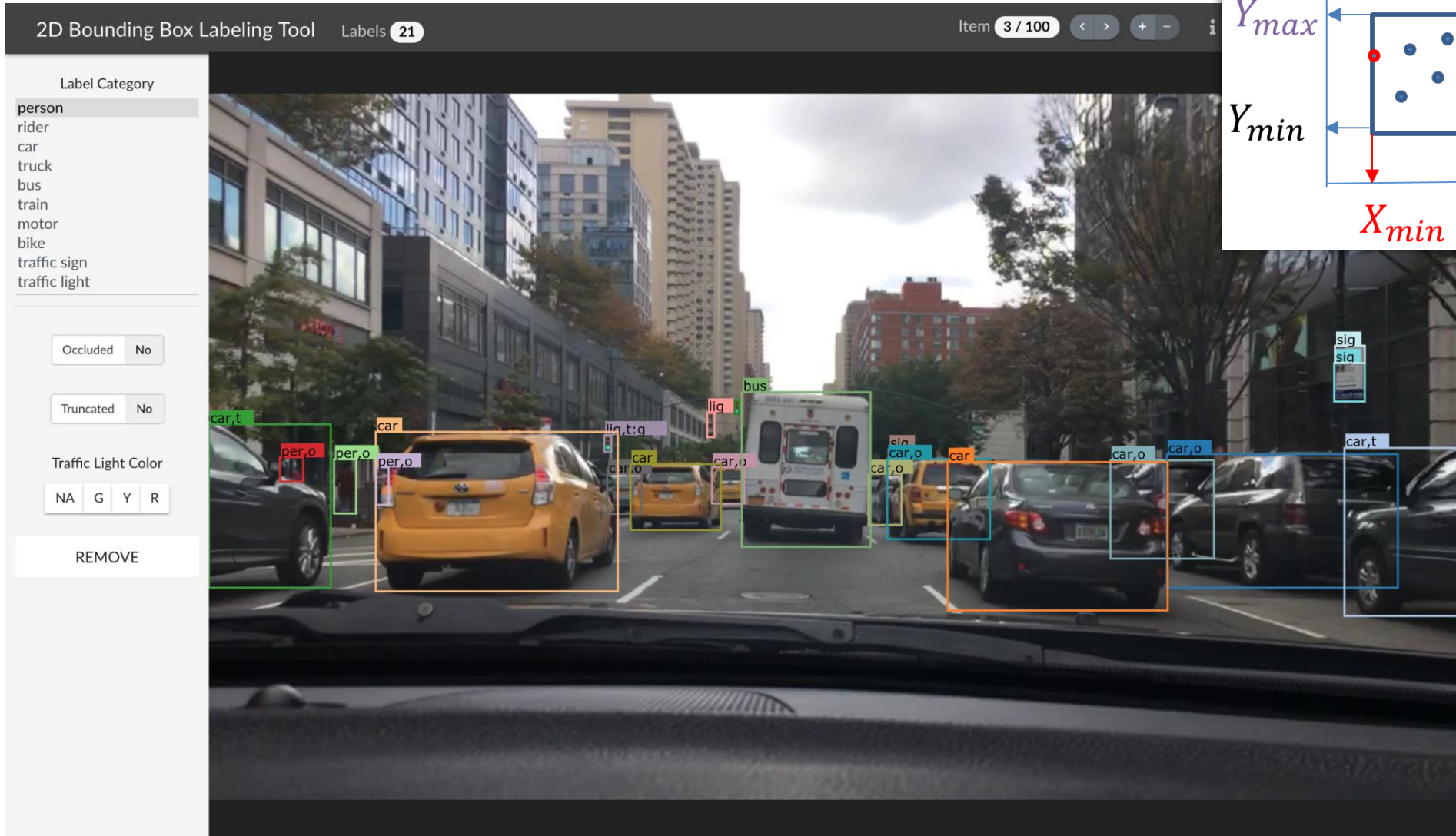
Cache demand

Content's Popularity

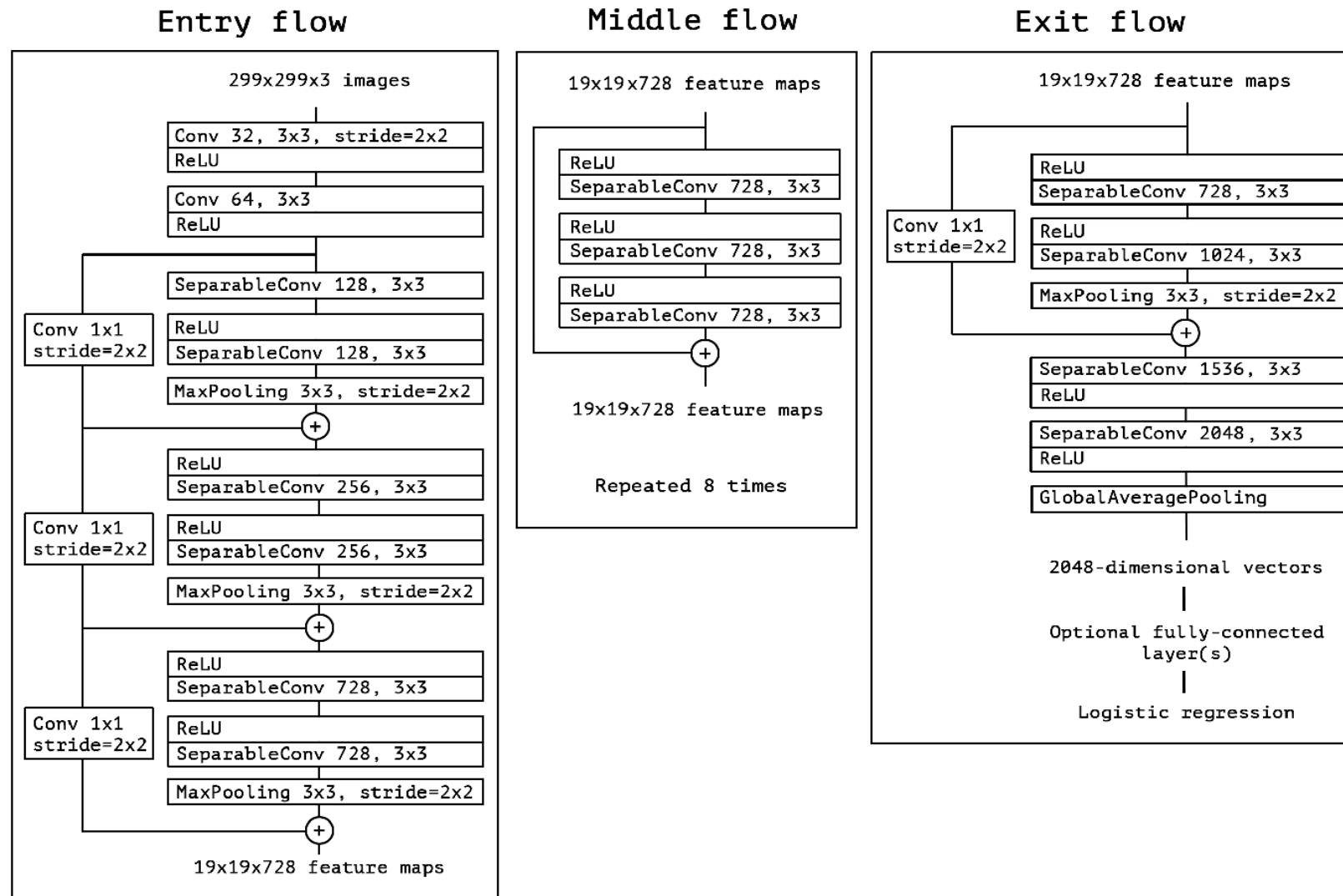


License Plate Detection System

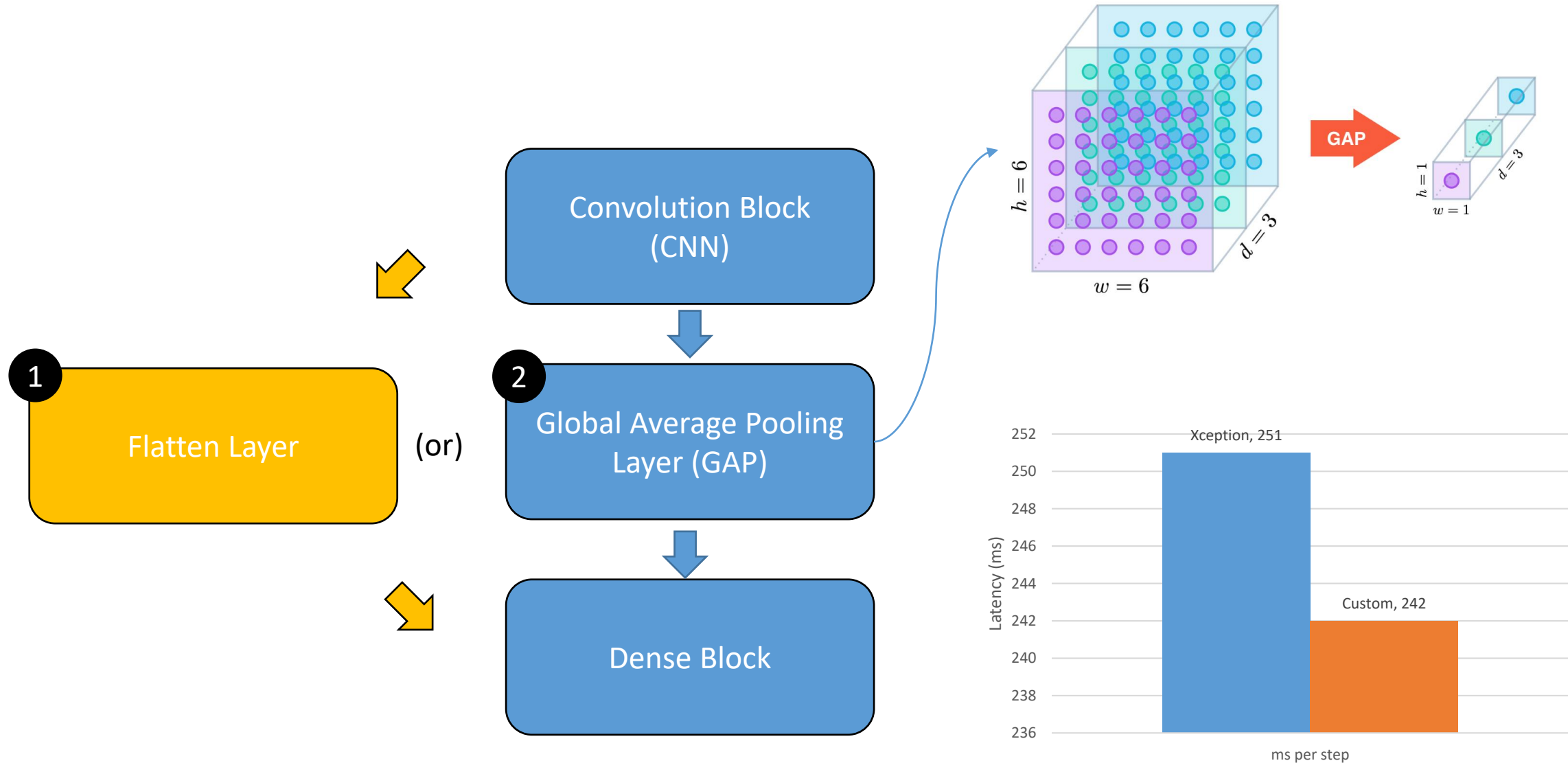
Bounding Box



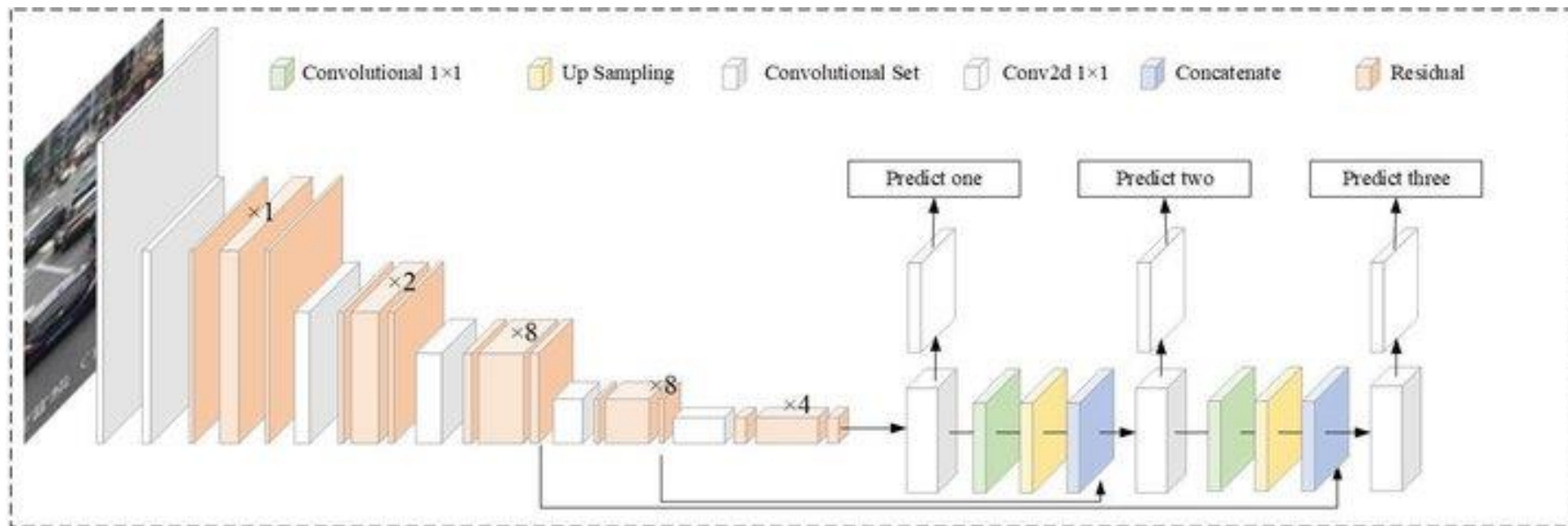
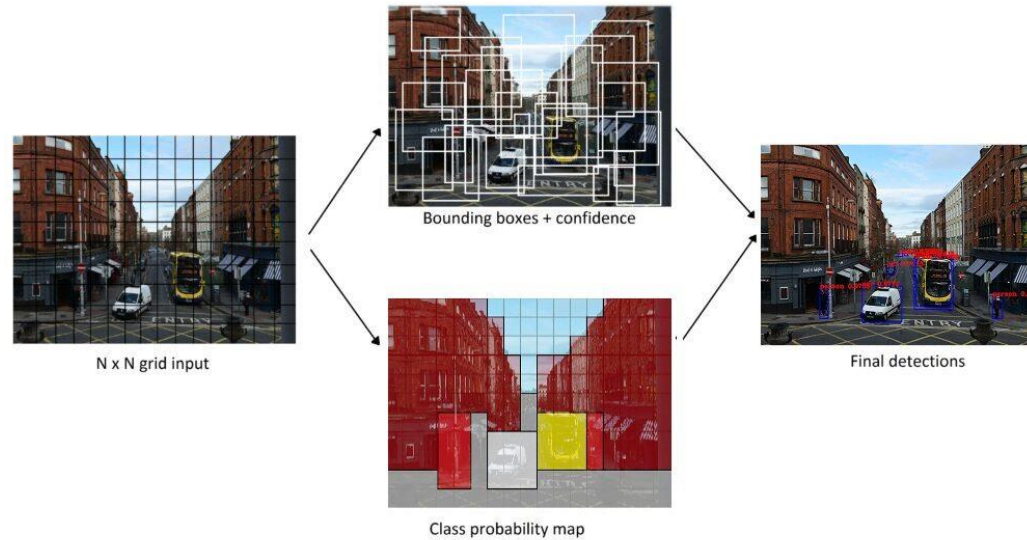
License Plate Location Detection - Xception Model



License Plate Location Detection - Custom Model

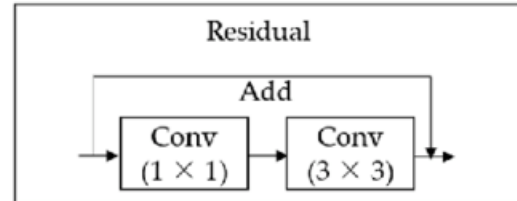
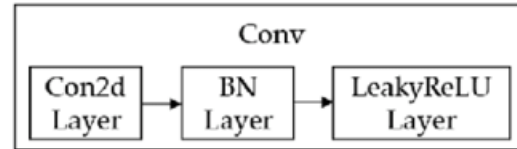


License Plate Location Detection - Yolo Model



License Plate Location Detection - Yolo Model

Layer	Filters size	Repeat	Output size
Image			416 × 416
Conv	32 3 × 3/1	1	416 × 416
Conv	64 3 × 3/2	1	208 × 208
Conv	32 1 × 1/1	[Conv Residual] × 1	208 × 208
Conv	64 3 × 3/1		208 × 208
Residual			208 × 208
Conv	128 3 × 3/2	1	104 × 104
Conv	64 1 × 1/1	[Conv Residual] × 2	104 × 104
Conv	128 3 × 3/1		104 × 104
Residual			104 × 104
Conv	256 3 × 3/2	1	52 × 52
Conv	128 1 × 1/1	[Conv Residual] × 8	52 × 52
Conv	256 3 × 3/1		52 × 52
Residual			52 × 52
Conv	512 3 × 3/2	1	26 × 26
Conv	256 1 × 1/1	[Conv Residual] × 8	26 × 26
Conv	512 3 × 3/1		26 × 26
Residual			26 × 26
Conv	1024 3 × 3/2	1	13 × 13
Conv	512 1 × 1/1	[Conv Residual] × 4	13 × 13
Conv	1024 3 × 3/1		13 × 13
Residual			13 × 13

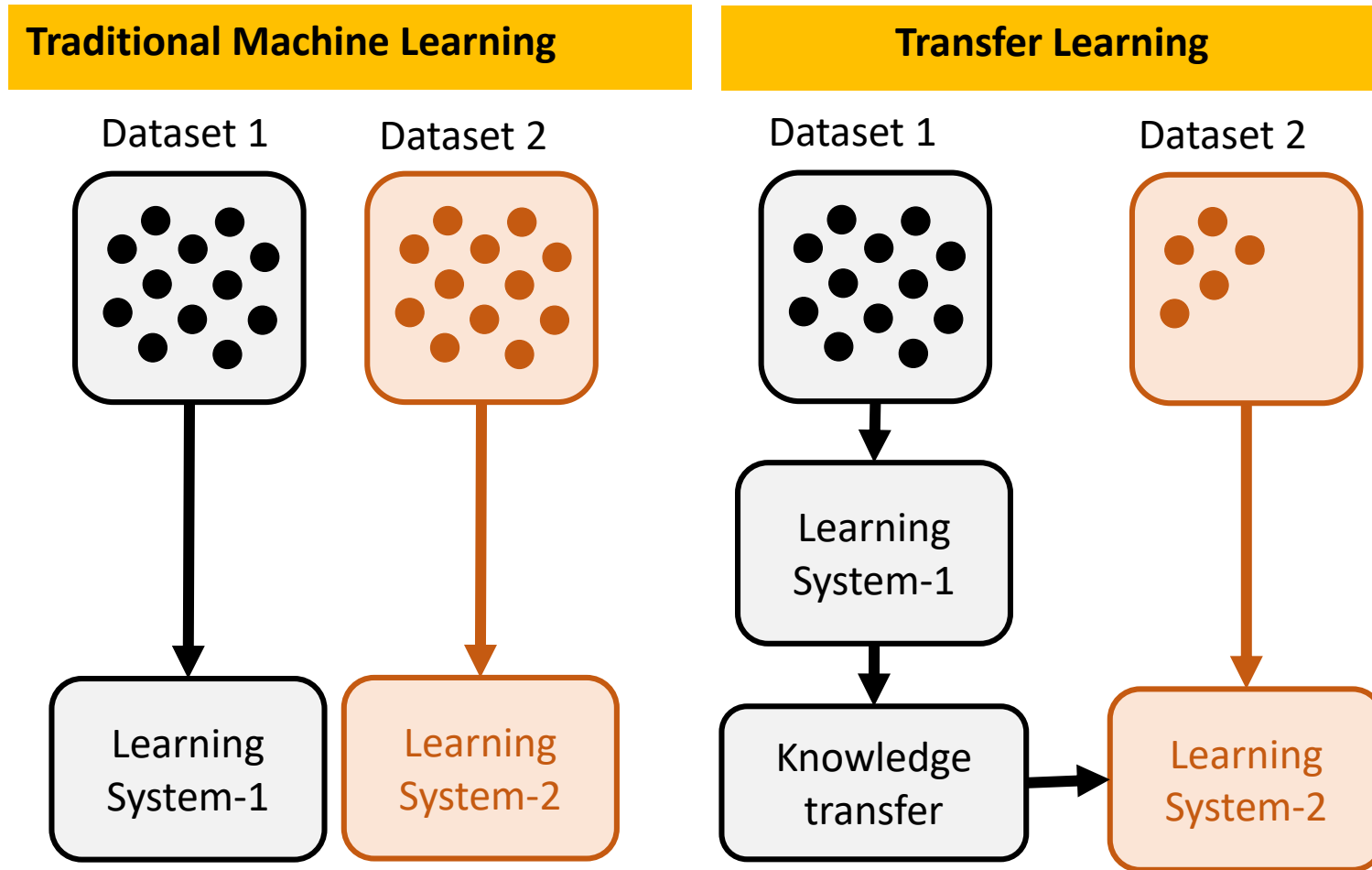


Darknet-52

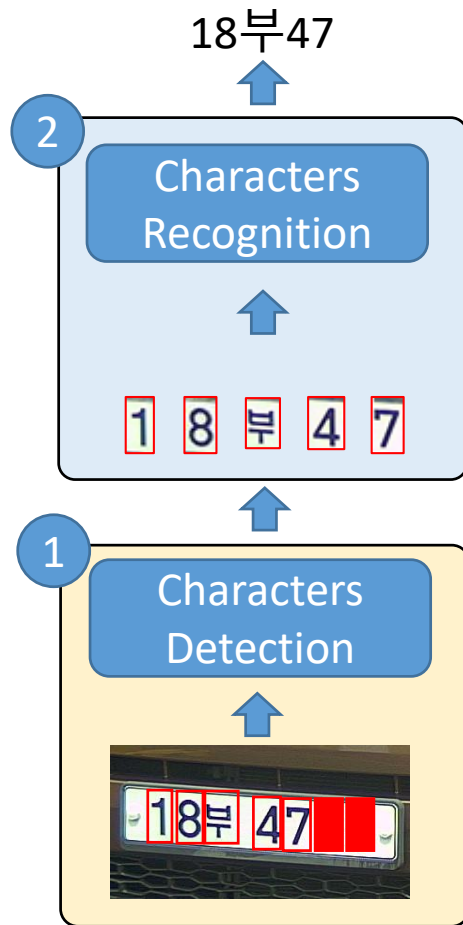
Type	Filters	Size/Stride	Output
Convolutional	32	3 × 3	224 × 224
Maxpool		2 × 2/2	112 × 112
Convolutional	64	3 × 3	112 × 112
Maxpool		2 × 2/2	56 × 56
Convolutional	128	3 × 3	56 × 56
Convolutional	64	1 × 1	56 × 56
Convolutional	128	3 × 3	56 × 56
Maxpool		2 × 2/2	28 × 28
Convolutional	256	3 × 3	28 × 28
Convolutional	128	1 × 1	28 × 28
Convolutional	256	3 × 3	28 × 28
Maxpool		2 × 2/2	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Maxpool		2 × 2/2	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	1000	1 × 1	7 × 7
Avgpool		Global	1000
Softmax			

Darknet-19

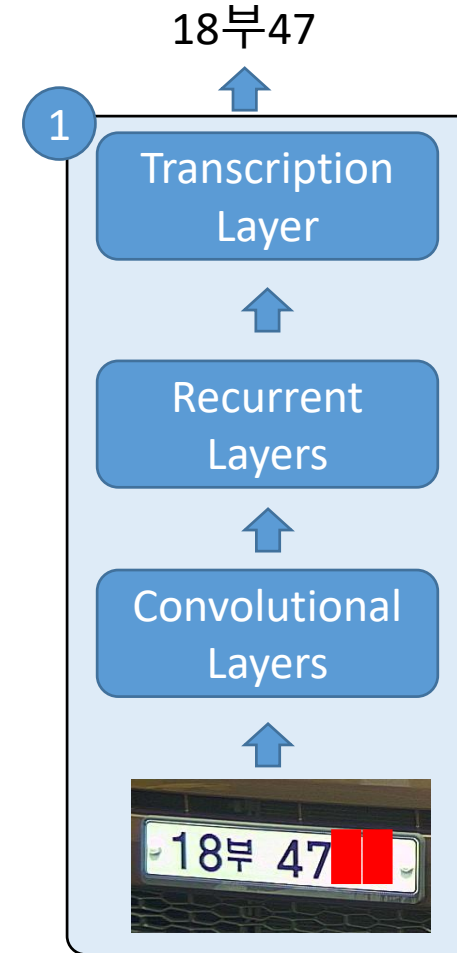
License Plate Location Detection - Yolo Model



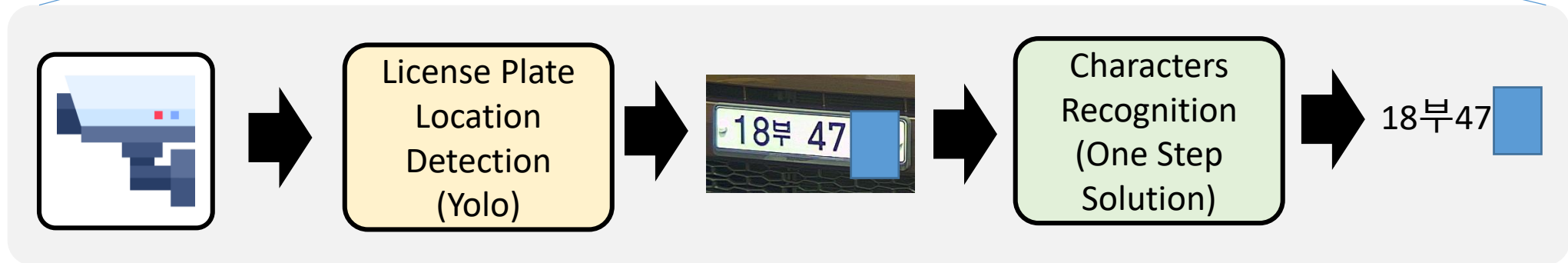
Characters Recognition



Two Steps Solution



One Step Solution



License Plate Detection System

- Previously, powerful AI apps required large, expensive data center-class systems to operate.
- But edge computing devices can reside anywhere, as demonstrated in the above use cases.
- AI at the edge offers endless opportunities that can help society in ways never before imagined.
- The use of light-weight edge intelligence will play a vital role for enabling a variety of applications in 5G and beyond wireless networks.

Thank You!

Q & A