

MLBlock

Introduction to Deep Learning

หัวข้อบรรยาย

- สอน AI ในสามชั่วโมง
- Artificial Intelligence, Machine Learning, Deep Learning
- สอน AI ให้ทำงานที่ต้องการ
- Real-time AI
- การเชื่อมต่อกับกล้องภายนอก, App อื่นๆ, โทรศัพท์มือถือ
- การเชื่อมต่อกับ Microcontroller ผ่าน NetPie, MQTT, Serial Port
- การประยุกต์ใช้ AI ในโครงการต่างๆ

MLBlock

- Visual Tools for Learning and Apply Deep Learning
- Based on TensorFlow.js, ML5 and P5
- Developed by Thai Developers with cooperation and endorsing by Nectec
 - Working with KidBright and NetPie
- Free and open source (MIT license)

ปัญญาประดิษฐ์ (Artificial Intelligence)

- ระบบที่สร้างขึ้นเพื่อเลียนแบบพฤติกรรมที่มี **ความฉลาด (Intelligence)**
- ระดับของปัญญาประดิษฐ์
 - Narrow AI
 - Generalized AI (Sentient Machine)
 - Super Intelligence

ความฉลาด (Intelligence)

- มีคุณสมบัติ 2 ประการ คือ
 - ทำงานได้เอง (Autonomous)
 - ปรับตัวหรือเปลี่ยนแปลงพฤติกรรมได้เมื่อมีอุปสรรคหรือปัญหาในการทำงาน (Adaptive)
- กระบวนการทำงานที่ต้องใช้ความฉลาดมี 3 ขั้นตอน
 - รับข้อมูล (Data Input)
 - ประมวลผลข้อมูล (Process)
 - แปลผล (Inference) และตัดสินใจ (Decision)
- ข้อมูล (Data) แบ่งเป็น 2 ประเภท
 - ข้อมูลเชิงปริมาณ (Quantitative Data)
 - ข้อมูลเชิงคุณภาพ (Qualitative Data) - รูป, รส, กลิ่น, เสียง, สัมผัส

การมองเห็นของสิ่งมีชีวิต (Biological Vision)

- Cambrian Explosion

- 500+ ล้านปีก่อน มีการเพิ่มขึ้นของจำพวก (Genera) ของสิ่งมีชีวิตมากขึ้นอย่างเห็นได้ชัด
- เริ่มต้นจาก Trilobite ที่มีวิวัฒนาการของอวัยวะรับแสง
 - หาอาหารได้ดีขึ้น
 - หลบหลีกภัยอันตรายและศัตรูได้รวดเร็ว
 - หาคู่ที่เหมาะสมได้ดี
- การมองเห็นเป็นปัจจัยสำคัญต่อการวิวัฒนาการของสัตว์

การศึกษาการทำงานของสมองในด้านการมองเห็น

- David Hubel และ Torsten Wiesel ศึกษาการทำงานของสมองในแมวเมื่อตอบสนองต่อการมองเห็น ในปี 1959
- พบว่าการทำงานของสมองแบ่งเป็นชั้นๆของเซลล์ประสาท (Neuron)
 - Simple Neuron ตอบสนองต่อรูปทรงง่ายๆ เช่น เส้นตรงในมุมใดมุมหนึ่ง
 - Complex Neuron รับข้อมูลเป็นชั้นๆต่อจาก Simple Neuron ทำหน้าที่ประมวลผลภาพที่ซับซ้อนขึ้น เช่น มุม หรือเส้นโค้ง
 - การทำงานเป็นชั้นๆของสมอง ทำให้สามารถแปลผลในเชิงนามธรรมของสิ่งที่มองเห็นได้

Perceptron

- Frank Rosenblatt จำลองการทำงานของเซลล์ประสาท (Neuron) ขึ้นในปี 1958 เรียกว่า “Perceptron”
- Marvin Minsky แสดงให้เห็นว่า Perceptron ไม่สามารถแก้ไขปัญหา XOR ได้ ทำให้การศึกษาด้าน AI แบบ Artificial Neuron เลื่อมความนิยม (AI Winter #1)

Neocognitron

- Kuniyuki Fukushima สร้างเครื่องจักรเลียนแบบการทำงานสมองจากผลงานวิจัยใน
แนวของ Hubel และ Wiesel ขึ้นในปี 1970

LeNet-5

- Yann LeCun และ Yoshua Bengio เสนอผลงานวิจัย LeNet ในปี 1989 ซึ่งมีการเรียนรู้โดยใช้เทคนิค Back-propagation (Backprop)
- LeNet-5 ถูกใช้โดยไปรษณีย์สหรัฐ (US Postal Service) เพื่ออ่านรหัสไปรษณีย์ ในปี 1998
- ใช้ฐานข้อมูล MNIST ในการสอนระบบ
- ด้วยข้อจำกัดด้านพลังการคำนวณของคอมพิวเตอร์จึงไม่สามารถ scale ไปสู่งานที่ใหญ่กว่านั้นได้ Neural Network จึงเสื่อมความนิยมไปอีกครั้ง (AI Winter #2)

Traditional Machine Learning

- Paul Viola และ Michael Jones สร้าง Machine Learning algorithm สำหรับตรวจจับใบหน้าแบบ real-time ได้สำเร็จ ในปี 2000
- ใช้ในระบบ autofocus ของกล้อง Fujifilm

ความล้มเหลวของ Deep Learning (AI Winters)

- AI Winter #1
 - ขาดการพัฒนา algorithm ที่เหมาะสม และด่วนตัดสินใจว่าใช้ไม่ได้
 - Multi-layer neural network
 - Backpropagation
- AI Winter #2
 - จำนวนข้อมูลที่ใช้ในการสอนมีไม่มากพอ (MNIST เป็นเพียงฐานข้อมูลของรูปภาพตัวเลข)
 - กำลังในการคำนวณของคอมพิวเตอร์ไม่พอ

สิ่งจำเป็นของ Deep Learning

- Algorithm
- Data
- Computing Power

ImageNet

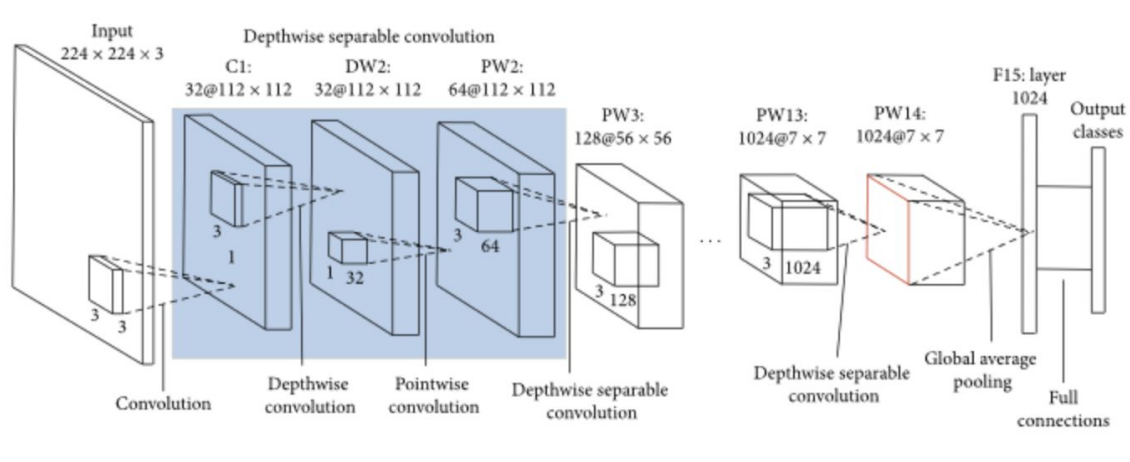
- Fei-Fei Li สร้างฐานข้อมูลภาพขนาดใหญ่ขึ้นในปี 2006 เรียกว่า ImageNet
- ประกอบด้วยรูปภาพพร้อมคำบรรยายสิ่งที่อยู่ในภาพจำนวน 14 ล้านภาพ แบ่งเป็น 22,000 ประเภท (category) ที่จัดเป็นลำดับความสัมพันธ์ (Heirachial database)
- จัดการแข่งขัน ILSVRC ให้นักวิทยาศาสตร์จำแนกภาพโดยใช้ฐานข้อมูล 1.4 ล้านภาพ แยกภาพเป็น 1000 จำพวก

AlexNet

- เข้าแข่งขัน ILSVRC ในปี 2012 โดย Alex Krizhevsky และ Ilya Sutskever จากมหาวิทยาลัย Toronto ภายใต้การนำของ Geoffrey Hinton
- ใช้ GPU ในการคำนวณ
- นำโลกเข้าสู่ยุคของ Deep Learning

MobileNet

- พัฒนาโดย Andrew G. Howard จาก Google ในปี 2017
- เป็น Open Source
- มีขนาดเล็ก และทำงานเร็ว เหมาะกับเครื่องคอมพิวเตอร์ส่วนบุคคลและสมาร์ทโฟน



TensorFlow

- Google's Open-source Machine Learning Library
- เขียนด้วยภาษา C++
- นิยมใช้ผ่านภาษา Python

TensorFlow.js

- Machine Learning Library สำหรับ JavaScript ที่พัฒนาขึ้นให้เหมือนการทำงานของ TensorFlow
- ใช้ใน Web Browser ได้
- ใช้การคำนวณโดยใช้ GPU ของการ์ดจอผ่าน WebGL
- ใช้ในฝั่ง Server ได้โดยรันบน Node.js

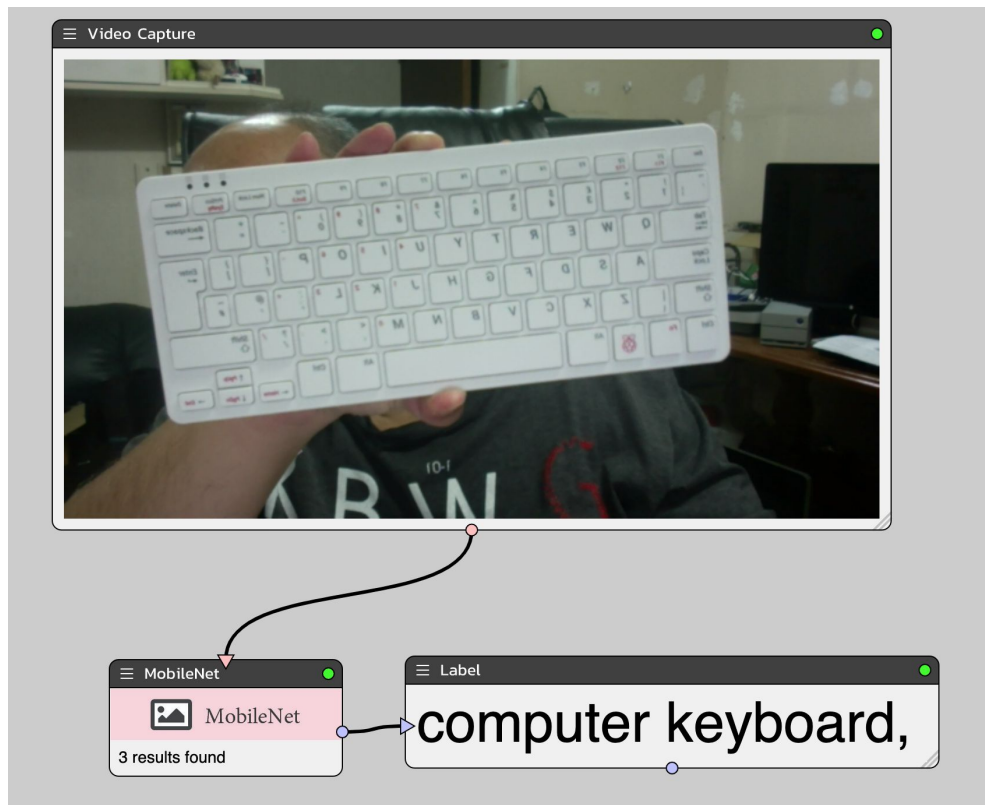
On-device Machine Learning

- Introduced by Google in Google I/O 2021
- แนวโน้มในการใช้ AI บนเครื่องฝั่ง Client
 - Android
 - iOS
 - Web
- ข้อดีของ On-device
 - Camera and sensors usage
 - Low Latency
 - Keep data on device
 - Works offline
 - No cost

MLBlock

- Visual Tools for Learning and Apply Deep Learning
- Based on TensorFlow.js, ML5 and P5
- Developed by Thai Developers with cooperation and endorsing by Nectec
 - Working with KidBright and NetPie
- Free and open source (MIT license)

MobileNet in MLBlock

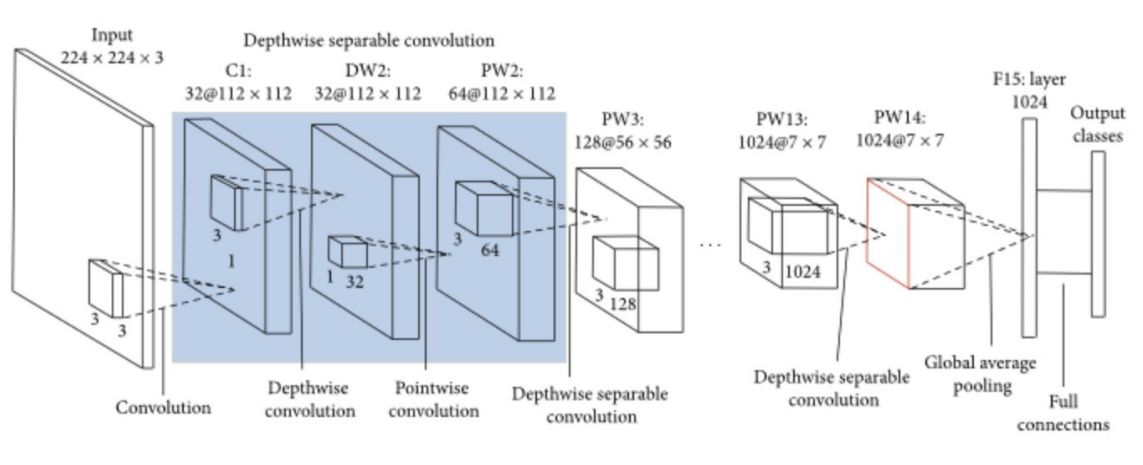


Transfer Learning

- การนำเอา Pre-trained Model มาสอนให้ทำงานตามที่ต้องการ
- เช่น การนำเอา MobileNet มาสอนให้รู้จักภาพใหม่

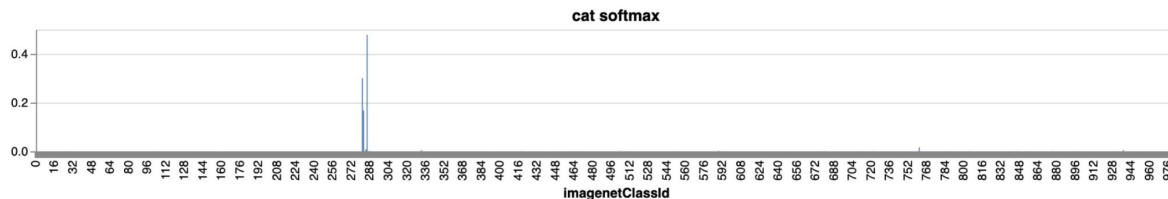
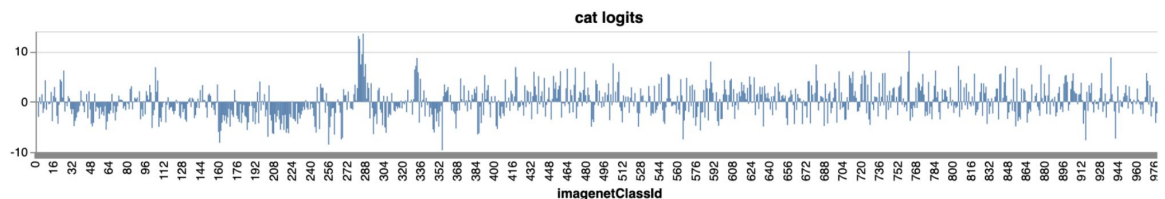
สถาปัตยกรรมของ MobileNet

- Depthwise Separable Convolutional Neural Network



สถาปัตยกรรมของ MobileNet

- Last Layer is Softmax to classify 1000 categories of images
- Before that is unnormalized prediction vector (Feature Vector) called “Logits”



MobileNet Transfer Learning

- Remove Softmax Layer
- Transfer Learning
 - Train new Softmax Layer
 - Classification
 - Regression
 - Real-time Inference by using KNN
 - Creating Neural Network

Demo

MobileNet Transfer Learning

- Remove Softmax Layer
- Transfer Learning
 - Train new Softmax Layer
 - Classification
 - Regression
 - Real-time Inference by using KNN

Demo

MobileNet Transfer Learning

- Remove Softmax Layer
- Transfer Learning
 - Train new Softmax Layer
 - Classification
 - Regression
 - Real-time Inference by using KNN
 - Creating Neural Network

Demo

Other Model Transfer Learning

- Using Neural Network to Transfer Learning Pre-trained model
 - PoseNet
 - HandPose
 - FaceAPI

Demo

Apply Deep Learning

- Turn Deep Learning to Decision and Control
 - IoT devices
 - Security
 - Automation Access Control
 - etc.
- Apply What you learn
 - Kidbright
 - NetPie
 - MLBlock

Game Time