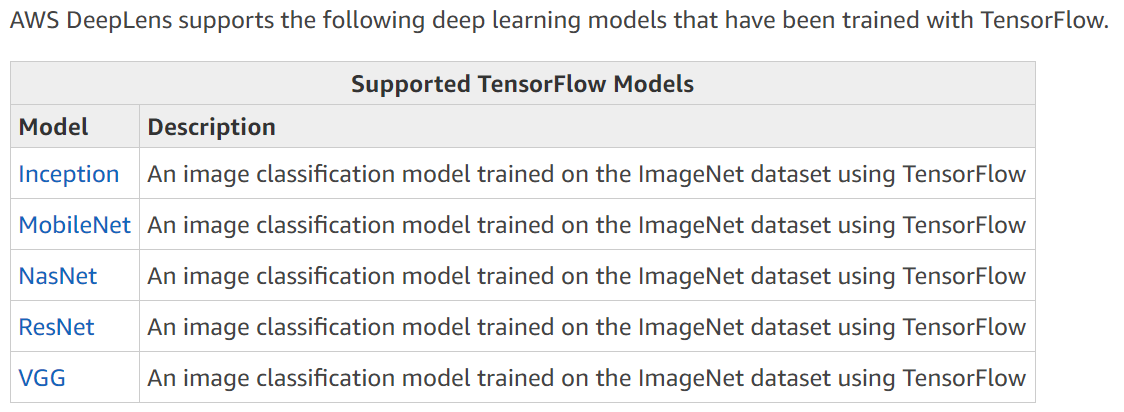
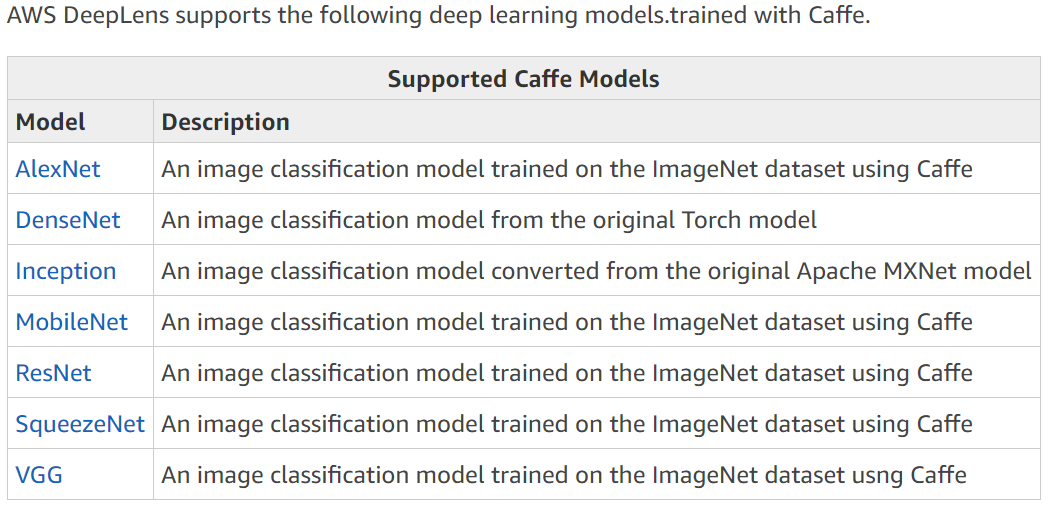
# **AWS Deep Lens**

# What is AWS Deep Lens?

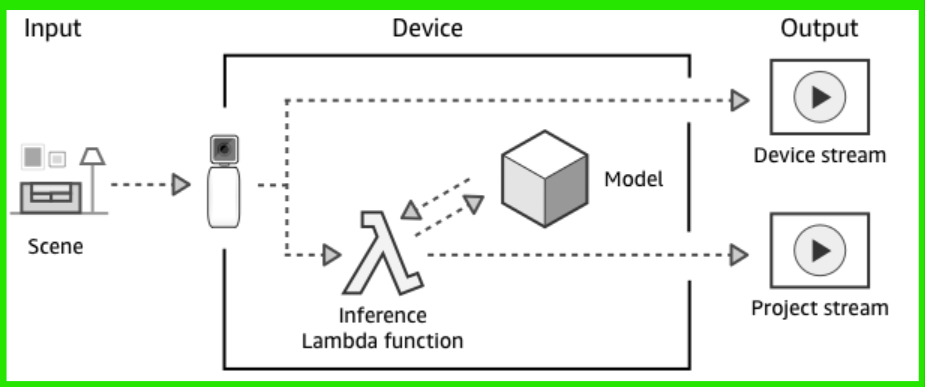
* AWS Deep Lens is a deep learning-enabled video camera
* **Device Version :** Launched in June 14, 2018
* **Hardware :**
* Camera status: when it is on, it indicates that a project is deployed successfully to the device and the project is running. Otherwise, the LED light is off



* Supported ML Frameworks: Deep learning frameworks, including Caffe, MX Net and TensorFlow

* **Project Work Flow :**
* When turned on, the AWS Deep Lens captures a video stream.
* Your AWS Deep Lens produces two output streams:
  + Device stream—The video stream passed through without processing.
  + Project stream—The results of the model's processing video frames
* The Inference Lambda function receives unprocessed video frames
* The Inference Lambda function passes the unprocessed frames to the project's deep learning model, where they are processed.
* The Inference Lambda function receives the processed frames from the model and passes the processed frames on in the project stream



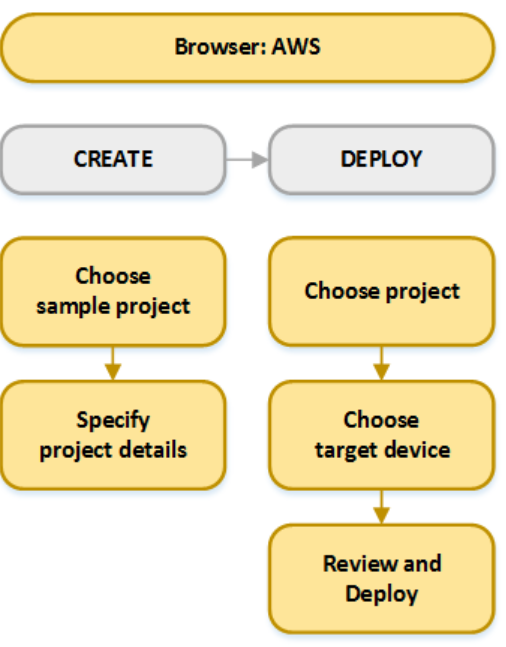
# Using AWS Deep Lens

Register you device:

* URL: <https://aws.amazon.com/deeplens/>

# Using AWS Deep Lens Project

* Create an AWS Deep Lens project on the AWS Cloud and deploy it to run on the device



# View Video Streams from AWS Deep Lens

To view project or live streams from the AWS Deep Lens 2019 Edition device in a browser, we need:

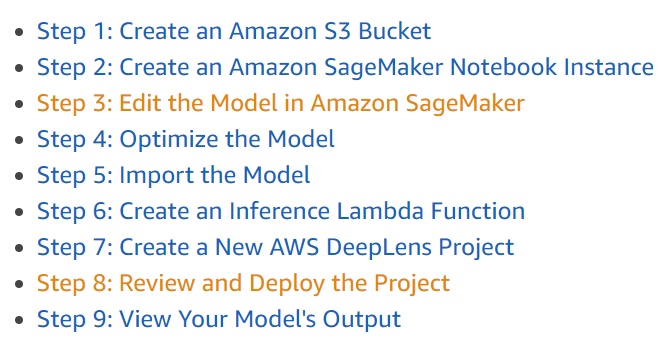
* self-signed streaming certificate downloaded
* uploaded to a supported browser you'll use

To view project stream, we will need an AWS Lambda function that interacts with the mjpeg stream on your device and the deep learning model

* <https://console.aws.amazon.com/lambda/home?region=us-east-1#/discover>

# Use Amazon Sage Maker to Provision a Pre-trained Model for a Sample Project

<https://console.aws.amazon.com/sagemaker/home?region=us-east-1#/landing>



# Import an Externally Trained Model

<https://us-east-1.console.aws.amazon.com/deeplens/home?region=us-east-1#models/import>.