

# メディアアート・プログラミング

## ml5.js 実践 - 転移学習 1 : 特徴抽出による画像のクラス分け

2019年10月23日  
前橋工科大学総合デザイン工学科  
田所 淳

# 今日の内容

---

- ▶ 転移学習をプログラミングしてみる
- ▶ 転移学習とは？
- ▶ 転移学習を活用したサンプル
  - ▶ 特徴抽出による画像のクラス分け

# 転移学習 (Transfer Learning) とは？



# 転移学習 (Transfer Learning) とは？

---

- ▶ これまでやってきたように、機械学習では膨大なデータセットが必要
- ▶ 例えば、ImageNetは14,197,122のラベル付けされた画像のデータセット
- ▶ これを個人で用意するのは困難





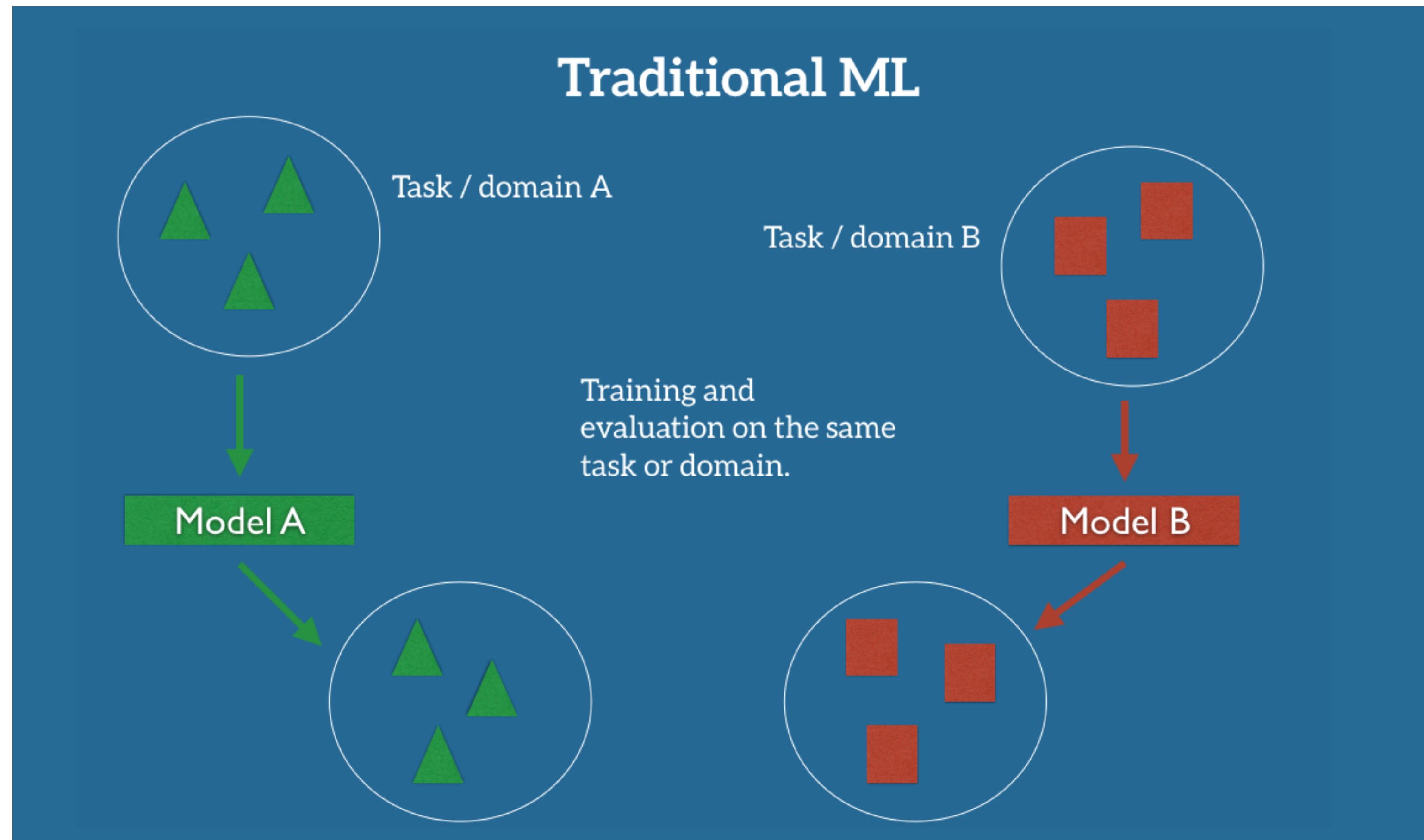
# 転移学習 (Transfer Learning) とは？

---

- ▶ 膨大なデータを用意しなくても効率的に学習を行える手法 → 転移学習
- ▶ 転移学習 (Transfer Learning)
- ▶ ある領域で学習させたモデルを、別の領域に適応させる技術

# 転移学習 (Transfer Learning) とは？

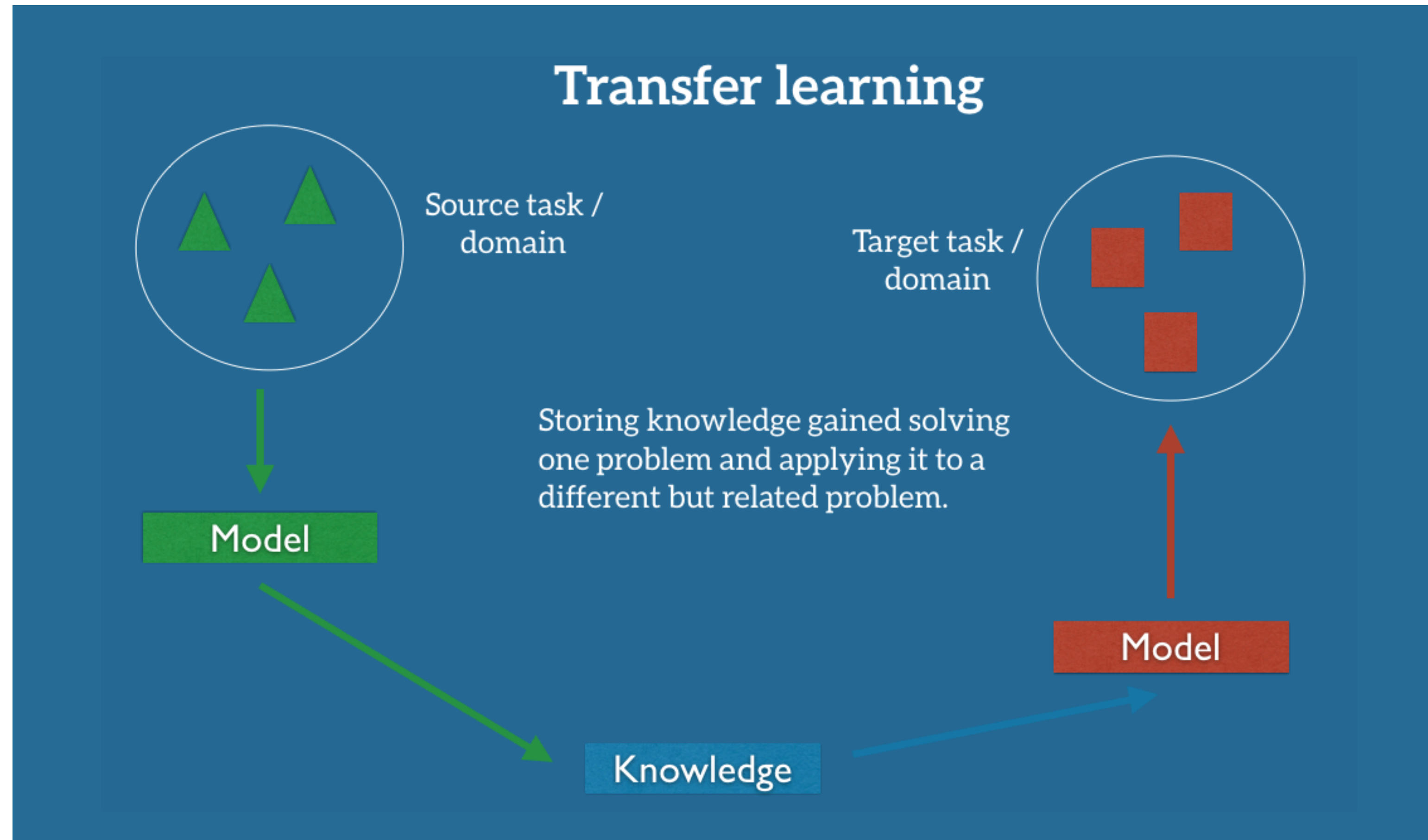
- ▶ 伝統的な機械学習の手法



Transfer Learning - Machine Learning's Next Frontier より

# 転移学習 (Transfer Learning) とは？

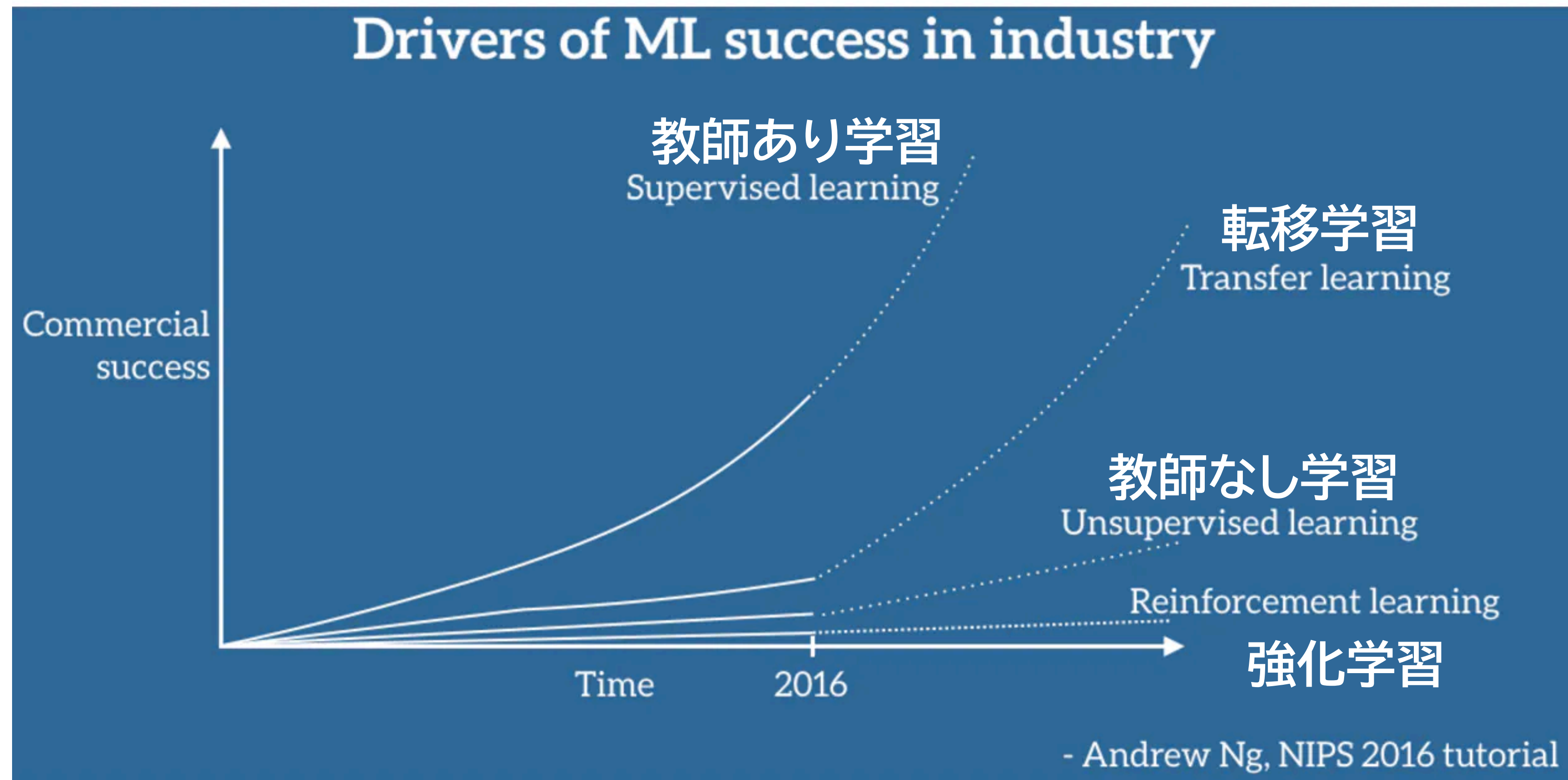
## ▶ 転移学習



Transfer Learning - Machine Learning's Next Frontier より

# 転移学習 (Transfer Learning) とは？

- ▶ 転移学習 - 今後の発展が期待される手法

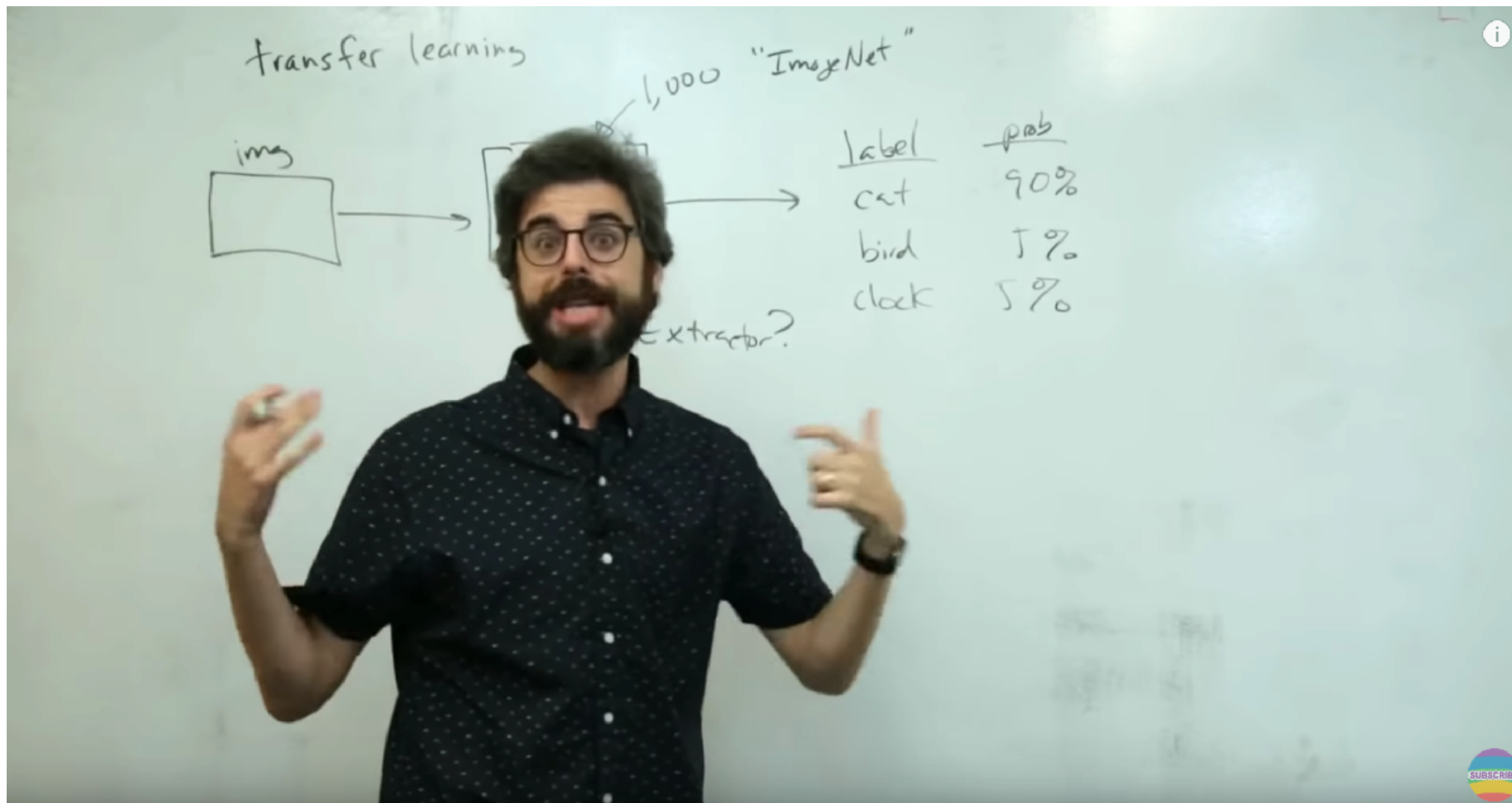


[Transfer Learning - Machine Learning's Next Frontier より](#)



# 転移学習 (Transfer Learning) とは？

- ▶ 今回も、Dan Shiffman先生の解説を観てみる
- ▶ ml5.js: Transfer Learning with Feature Extractor
- ▶ <https://www.youtube.com/watch?v=kRpZ5OqUY6Y>



ml5.js 転移学習による特徴抽出でイメージのクラス分け

# ml5.js 転移学習による特徴抽出でイメージのクラス分け

---

- ▶ ここまでの理解を踏まえて、ml5.jsで転移学習による特徴抽出をプログラミング
- ▶ 今回はイメージのクラス分けを行う



# ml5.js 転移学習による特徴抽出でイメージのクラス分け

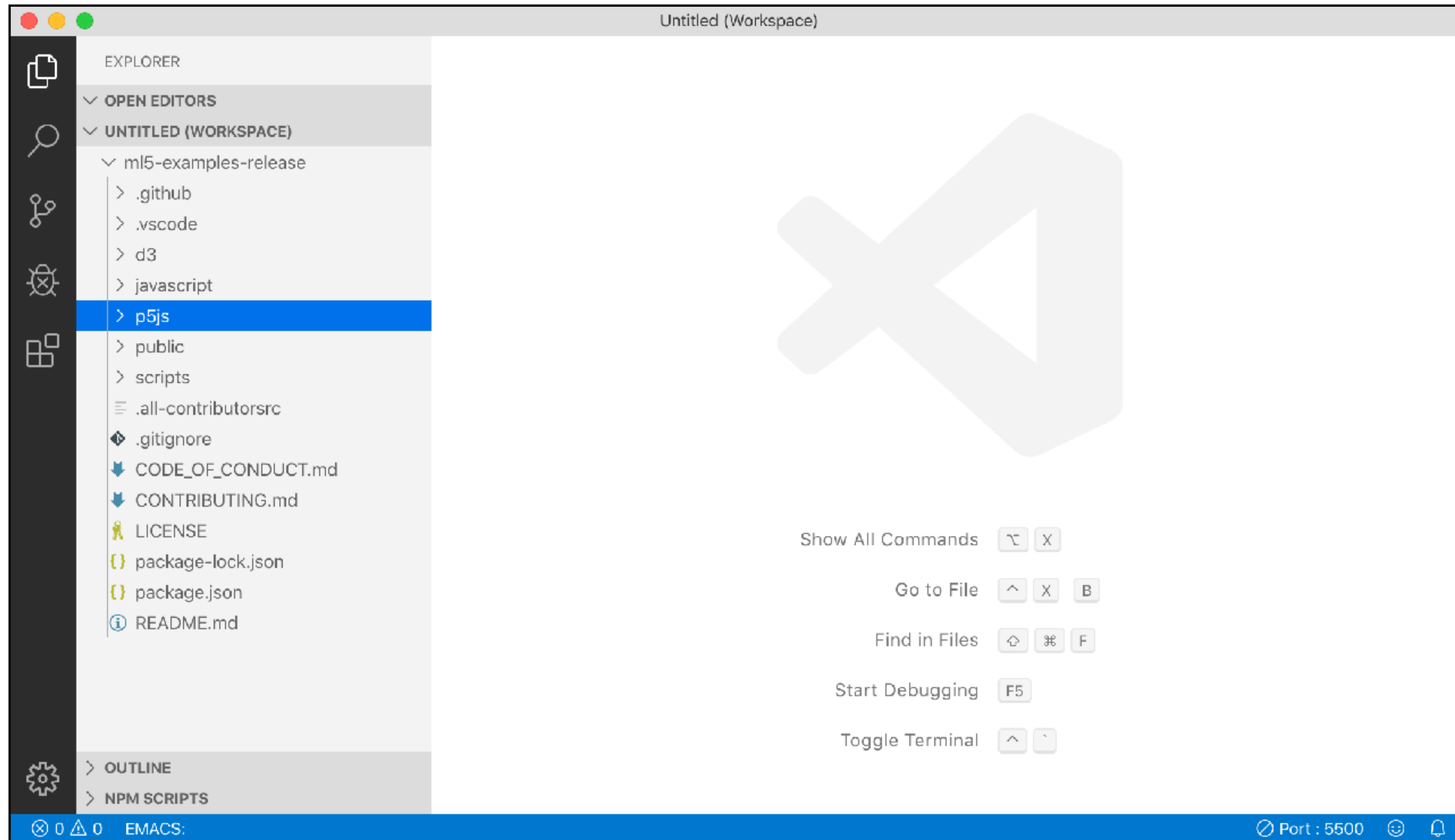
- ▶ ベースとなるプログラムを、githubからダウンロード
- ▶ ml5-examples
- ▶ <https://github.com/ml5js/ml5-examples>

The screenshot shows the GitHub repository page for `ml5js / ml5-examples`. At the top, it displays the repository name and statistics: 35 Watchers, 435 Stars, and 275 Forks. Below this is a navigation bar with links to Code, Issues (36), Pull requests (4), Projects (0), Wiki, Security, and Insights. The main content area describes the repository as "A collection of ml5.js examples" and provides a link to <https://ml5js.github.io/ml5-examples/...>. It also shows repository statistics: 371 commits, 12 branches, 5 releases, 17 contributors, and the MIT license. A progress bar is visible below these statistics. The "Branch: release" dropdown is set to "release", and there are buttons for "New pull request", "Create new file", "Upload files", "Find file", and "Clone or download". The file list shows the following structure:

File/Folder	Commit Message	Commit Hash	Time Ago
<code>.github</code>	ml5 v0.4.0 examples (#216)	09f674c	5 days ago
<code>.vscode</code>	transferring spelling		last year
<code>d3/KMeans/KMeans_GaussianClust...</code>	ml5 v0.4.0 examples (#216)		5 days ago
<code>javascript</code>	ml5 v0.4.0 examples (#216)		5 days ago
<code>p5js</code>	ml5 v0.4.0 examples (#216)		5 days ago
<code>public</code>	ml5 v0.4.0 examples (#216)		5 days ago
<code>scripts</code>	ml5 v0.4.0 examples (#216)		5 days ago

# ml5.js 転移学習による特徴抽出でイメージのクラス分け

- ▶ Visual Studio Codeで開いて、Live Serverを起動



# ml5.js 転移学習による特徴抽出でイメージのクラス分け

## ▶ サンプルのページ

### Image Classification using Feature Extractor with MobileNet



MobileNet Loaded!

Add Cat Image 0 Cat Images

Add Dog Image 0 Dog Images

Add Badger Image 0 Badger Images

Train

Start guessing!

Label: ...

confidence: ...

Save Model

| Load Model:

Choose Files

No file chosen



# ml5.js 転移学習による特徴抽出でイメージのクラス分け

---

- ▶ サンプルのページの使い方 例：「パー」、「グー」、「チョキ」を学習させてみる
- ▶ 1. 撮影
  - ▶ グーの手をカメラに写し「Add Cat Image」ボタンを押す（いろいろなアングルで）
    - ▶ 最低20枚以上
  - ▶ 同様に、チョキで「Add Dog Image」
  - ▶ さらに、パーで「Add Badger(アナグマ) Image」
- ▶ 2. 学習：「Train」ボタンを押す
- ▶ 3. クラス分け開始
  - ▶ 「Start Guessing」ボタンを押す
  - ▶ カメラの前で「グー」「チョキ」「パー」を写す
  - ▶ 結果が表示される（はず）

# ml5.js 転移学習による特徴抽出でイメージのクラス分け

---

- ▶ サンプルはWebのインターフェイス（ボタンなど）を使用していて地味…
- ▶ もっと見栄えを良く改造したい
- ▶ カメラの映像はフルスクリーンで
- ▶ ボタンではなく、キー操作で学習させていく
  - ▶ [a] ボタン：イメージ1（Cat）の追加
  - ▶ [s] ボタン：イメージ2（Dog）の追加
  - ▶ [d] ボタン：イメージ3（Badger）の追加
  - ▶ [f] ボタン：学習（Train）
  - ▶ [g] ボタン：クラス分け開始

# ml5.js 転移学習による特徴抽出でイメージのクラス分け

---

- ▶ まず、index.htmlを編集



# ml5.js 転移学習による特徴抽出でイメージのクラス分け

- ▶ index.html - body要素内は、sketch.jsのscript要素だけに

```
<html>
  <head>
    <meta charset="UTF-8" >
    <title>Image Classification using Feature Extraction with MobileNet. Built with p5.js</title>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/p5.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/addons/p5.dom.min.js"></script>
    <script src="https://unpkg.com/ml5@0.4.0/dist/ml5.min.js" type="text/javascript"></script>
    <style>
      body {
        margin:0; padding:0;
      }
    </style>
  </head>
  <body>
    <script src="sketch.js"></script>
  </body>
</html>
```

# ml5.js 転移学習による特徴抽出でイメージのクラス分け

---

- ▶ 次にsketch.jsを編集

# ml5.js 転移学習による特徴抽出でイメージのクラス分け

## ▶ sketch.js - 1 / 4

```
let featureExtractor;
let classifier;
let video;
let loss;
let dogImages = 0;
let catImages = 0;
let badgerImages = 0;
let status = ''; //現在の状態を左上に表示
let result = ''; //クラス分け結果を中央に表示

function setup() {
  //キャンバスを画面全体に
  createCanvas(windowWidth, windowHeight);
  video = createCapture(VIDEO);
  video.size(320, 240);
  video.hide();
  //学習設定
  featureExtractor = ml5.featureExtractor('MobileNet', modelReady);
  const options = { numLabels: 3 };
  classifier = featureExtractor.classification(video, options);
}
```



# ml5.js 転移学習による特徴抽出でイメージのクラス分け

## ▶ sketch.js - 2 / 4

```
function draw(){
  background(0);
  //ビデオ映像をフルスクリーン表示
  image(video, 0, 0, width, height);
  //学習させた画像の数を表示
  fill(255);
  textSize(12);
  textAlign(LEFT);
  text(status, 20, 20);
  text('Cat Images : ' + catImages, 20, 40);
  text('Dog Images : ' + dogImages, 20, 60);
  text('Badger Images : ' + badgerImages, 20, 80);
  //クラス分けした結果を表示
  fill(255, 255, 0);
  textSize(100);
  textAlign(CENTER);
  text(result, width/2, height/2);
}
```

# ml5.js 転移学習による特徴抽出でイメージのクラス分け

## ▶ sketch.js - 3 / 4

```
//ボタン操作からキー操作へ変更
function keyTyped() {
  if(key == 'a'){
    classifier.addImage('cat');
    catImages++;
  }
  if(key == 's'){
    classifier.addImage('dog');
    dogImages++;
  }
  if (key == 'd'){
    classifier.addImage('badger');
    badgerImages++;
  }
  if (key == 'f') {
    classifier.train(function(lossValue) {
      if (lossValue) {
        loss = lossValue;
        status = 'Loss: ' + loss;
      } else {
        status = 'Done Training! Final Loss: ' + loss;
      }
    });
  }
  if (key == 'g') {
    classify();
  }
}
```

# ml5.js 転移学習による特徴抽出でイメージのクラス分け

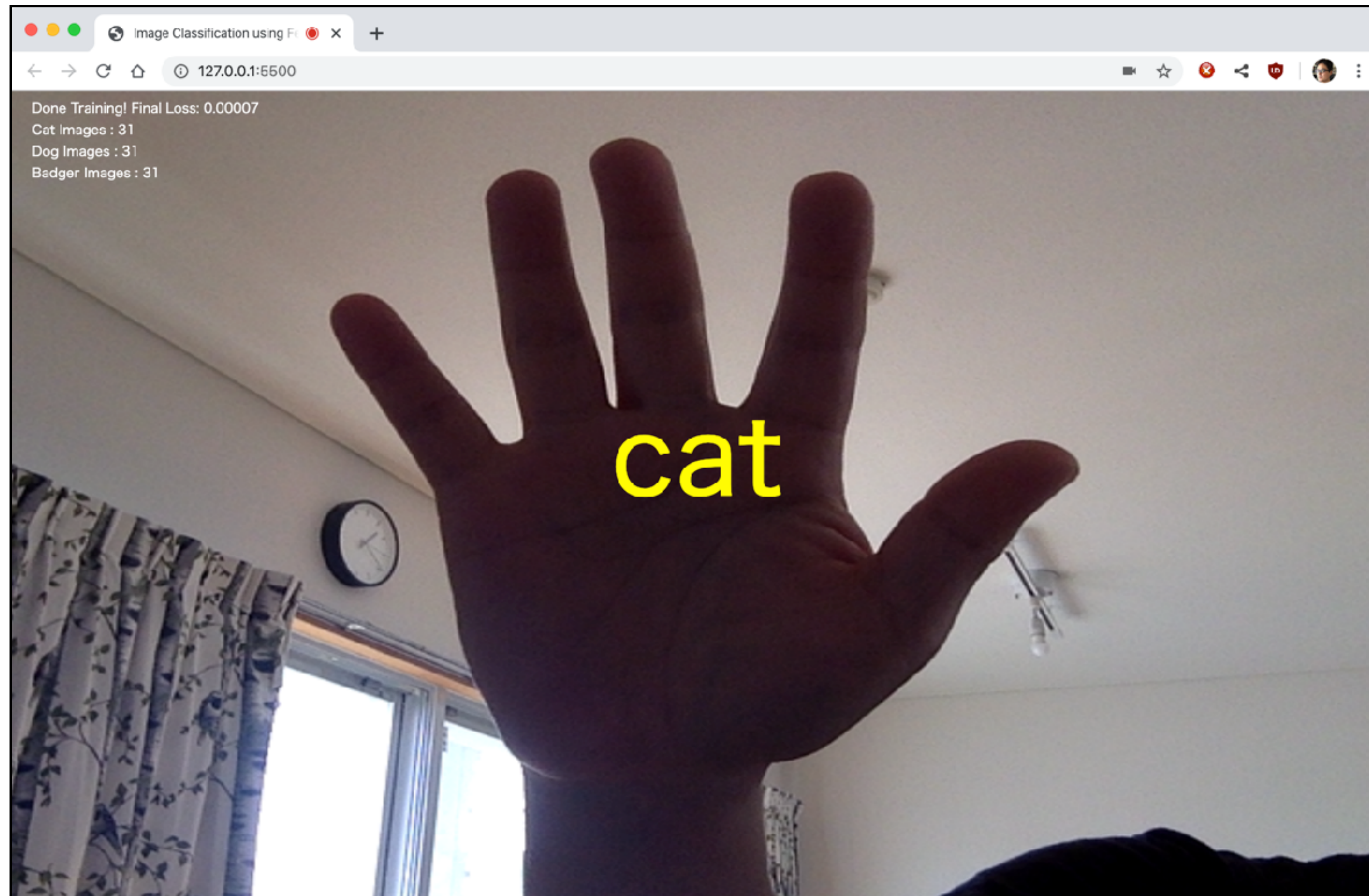
## ▶ sketch.js - 4 / 4

```
function modelReady() {  
  status = 'MobileNet Loaded!';  
}  
  
function classify() {  
  classifier.classify(gotResults);  
}  
  
function gotResults(err, results) {  
  //エラー表示  
  if (err) {  
    console.error(err);  
    status = err;  
  }  
  //クラス分けした結果を表示  
  if (results && results[0]) {  
    result = results[0].label;  
    classify();  
  }  
}
```



# ml5.js 転移学習による特徴抽出でイメージのクラス分け

▶ 完成!!





今日はここまで!