

AI技術を使った アプリを 作ってみた話

あべみ/Miyuki Kondo



ことの発端

> 2021年4月XX日

- 娘「この花なんていう花なの～？検索して調べられない？」



モチベーション 「子供と一緒に花画像検索アプリで遊びたい」

> モチベーション

- 画像検索、、、よく話には聞くけどどうなっているんだろう？
- AIアプリってどうやって世の中に公開するの？
- 車輪の再開発どんと来い、中身を理解&自分でAIサービス立ち上げられるようになるためにもちょっと作ってみよう！
- 持ち歩いているんな花で検索してみたい！

> 納期

- ~~花の種類が変わるまで（鬼畜）~~
- ~~GW中にタンポポがほぼほぼ綿毛に、ヒナゲシの花がほとんどなくなっ
て焦る~~

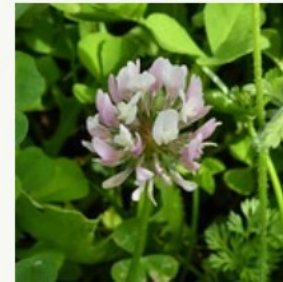
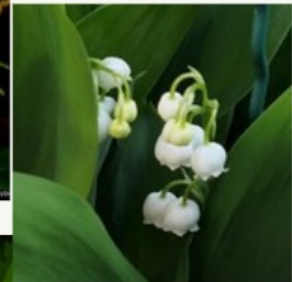


開発の流れ

- > データを集める
 - 今回は、娘にヒアリングした「画像検索したい10種類の花」を対象
- > モデルを作る
 - *ImageNetのFineTuning*
- > Webアプリとして公開
 - *Django*

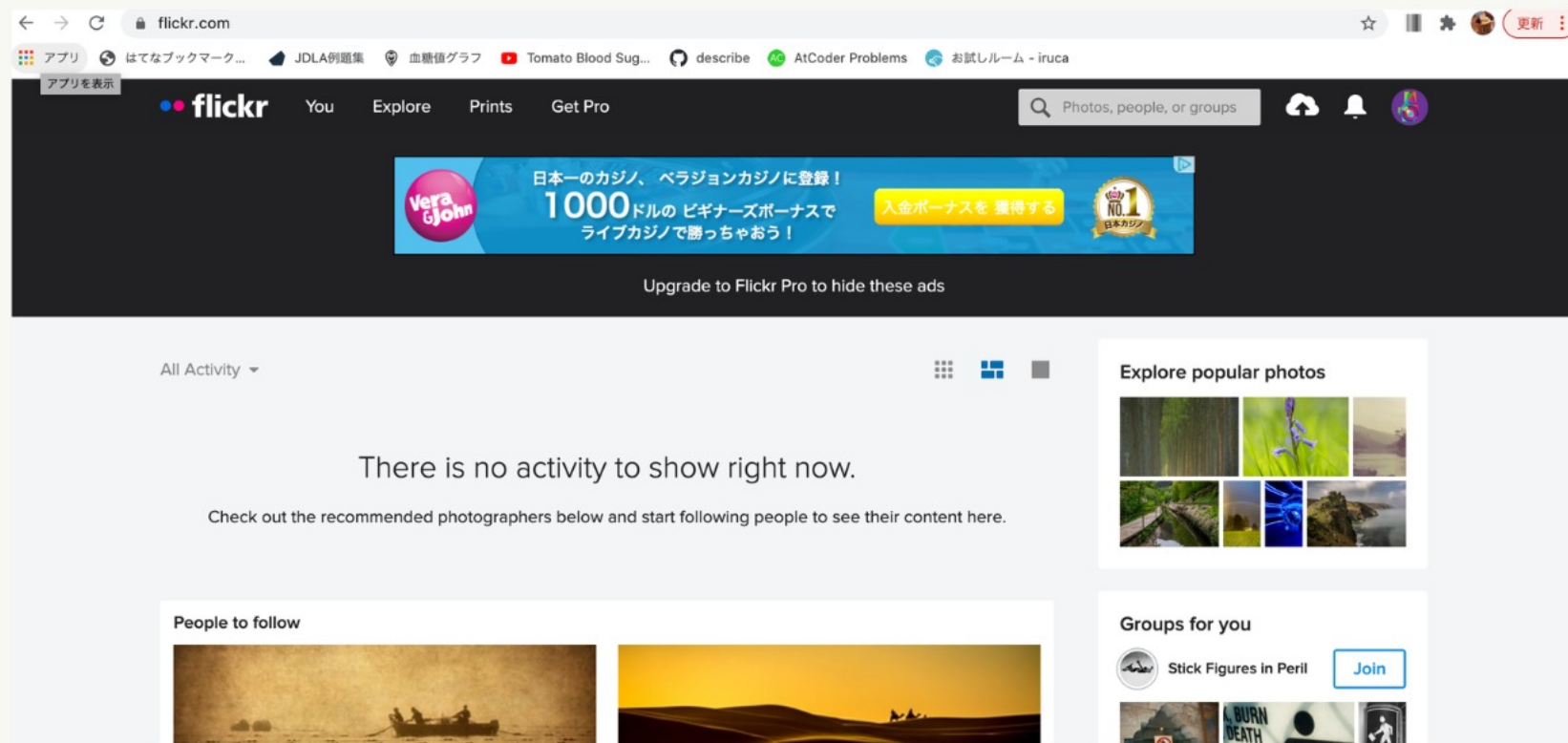
10種類の花

- > ヒナゲシ *rhoeas*
- > ムスカリ（ヒヤシンス） *hyacinth*
- > パンジー *pansy*
- > スミレ *viola mandshurica*
- > ツツジ *azalea*
- > スズラン *majalis*
- > タンポポ *dandelion* ※綿毛・黄色い花を混ぜる
- > シロツメクサ *trifolium repens* ※カタバミと間違いやすい
- > カタバミ *wood sorrel* ※シロツメクサと間違いやすい
- > サクラ *cherry blossom*



お花の画像の集め方

> download.pyを作ってFlicker APIを叩く





モデルを作る

> 教師データ

- Flickrで集めた、各カテゴリの200~300枚の画像
- 明らかに変な画像だけは目検で除く

> モデル

- ImageNetのResnetモデルを、Finetuningする

> 結果

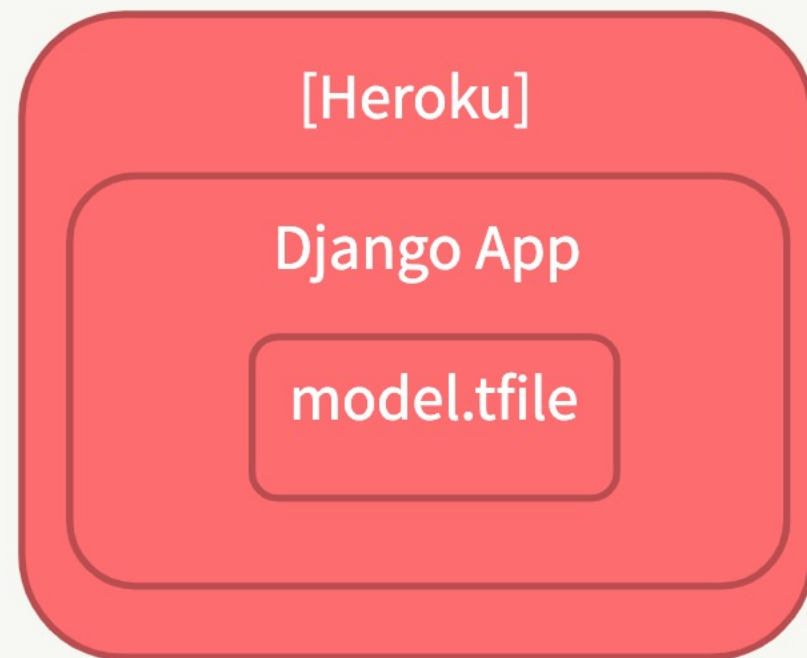
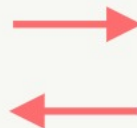
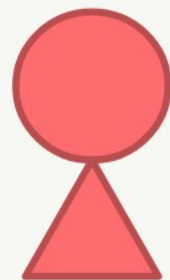
- 汎化性能で7割くらいのモデル※イマイチ、、、ただ今回はサービス化を優先

> モデルを保存

- h5ファイルにした後、tfliteファイルにする

デプロイ

- > やってみる
 - とりあえず、設定が簡単なのはHerokuなのでHerokuにトライ
- > Heroku + Djangoの構成（下記ファイルを用意してgit push heroku master）
 - *Requirement.txt*
 - > Pip install するmodule群
 - *Procfile*
 - > gunicornで起動する設定
 - *Runtime.txt*
 - > Pythonのver指定



詰まったこと

> herokuの容量制限 コンパイル後500MB制限

```
remote: -----> Compressing...
remote: !       Compiled slug size: 648M is too large (max is 500M).
remote: !       See: http://devcenter.heroku.com/articles/slug-size
remote:
remote: !       Push failed
remote: !
```

> TensorFlowのせいっぽい

```
remote: Collecting Django==3.2
remote:   Downloading Django-3.2-py3-none-any.whl (7.9 MB)
remote: Collecting grpcio==1.32.0
remote:   Downloading grpcio-1.32.0-cp38-cp38-manylinux2014_x86_64.whl (3.8 MB)
remote: Collecting h5py==2.10.0
remote:   Downloading h5py-2.10.0-cp38-cp38-manylinux1_x86_64.whl (2.9 MB)
remote: Collecting numpy==1.19.5
remote:   Downloading numpy-1.19.5-cp38-cp38-manylinux2010_x86_64.whl (14.9 MB)
remote: Collecting Pillow==8.2.0
remote:   Downloading Pillow-8.2.0-cp38-cp38-manylinux1_x86_64.whl (3.0 MB)
remote: Collecting protobuf==3.15.8
remote:   Downloading protobuf-3.15.8-cp38-cp38-manylinux1_x86_64.whl (1.0 MB)
remote: Collecting tensorboard==2.5.0
remote:   Downloading tensorboard-2.5.0-py3-none-any.whl (6.0 MB)
remote: Collecting tensorboard-data-server==0.6.0
remote:   Downloading tensorboard_data_server-0.6.0-py3-none-manylinux2010_x86_64.whl (3.9 MB)
remote: Collecting tensorboard-plugin-wit==1.8.0
remote:   Downloading tensorboard_plugin_wit-1.8.0-py3-none-any.whl (781 kB)
remote: Collecting tensorflow==2.4.1
remote:   Downloading tensorflow-2.4.1-cp38-cp38-manylinux2010_x86_64.whl (394.4 MB)
remote: Collecting tensorflow-estimator==2.4.0
remote:   Downloading tensorflow_estimator-2.4.0-py2.py3-none-any.whl (462 kB)
```

Tensorflow
394.4MB

対策１：とりあえずモデルを分離

- > TensorFlowを削除できないので、モデル（約100MB）を退避してみる

```
remote:          Proctile declares types -> web
remote:
remote: -----> Compressing...
remote:          Done: 486.5M
remote: -----> Launching...
remote: !      Warning: Your slug size (486 MB) exceeds our soft limit (300 MB)
           which may affect boot time.
```

- > モデルだけAmazonS3に置いてみる？
 - と思ったけど、なかなかうまくいかず
断念

セーフ！

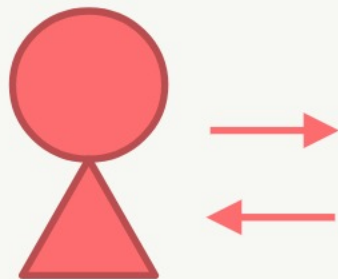
対策2：heroku諦めてまるっとAWSに

> 環境

- EC2 t2.micro (無料枠があるもの)
 - > Nginx + Gunicorn
- Elastic IP (EC2に紐付けたら無料のもの)

> その他TIPS

- Tensorflow
 - > バージョンを1.15.5にダウン
- Nginx
 - > 画像のUpload許可
サイズ設定



結果：AWSでできました

[画像アップロード画面]

画像判定AIアプリ

おはな判定アプリ

ファイルを選択して推定ボタンをクリックしてください

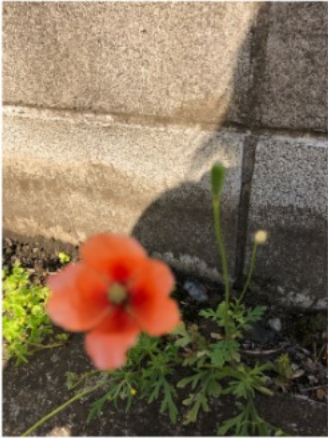
推定したいファイルを選択してください。

[予測結果画面]

画像判定AIアプリ

おはな判定結果

画像ファイル名 hinageshi_rina3.JPG

画像ファイル 

推定ラベル ヒナゲシ

学んだこと

> インフラ

- Herokuはデプロイ簡単だけどAIモデルのサービスはサイズオーバーで置けない
- AWSはHerokuよりは作業多いけどAIモデルを置ける（難しくはない）

> Django

- アプリ作りやすい

> AI

- Flickrで画像をいい感じに集められる
- ただ、画像にばらつきがあるのでもっと精度をあげるには学習に使う画像の精査がもっと必要そう
- 300枚はやっぱり少ないんだろうな、、、
- 類似画像検索でカテゴリを予想するのもありかも（AI Questで実施したファッションレコメンドと同じ手法）
- タンポポの綿毛と黄色い花はわけた方が良さそう

ご清聴ありがとうございました

> 参考資料

- **Udemy: 【画像判定AIアプリ開発パート2】 Django ・ TensorFlow ・ Python ・ 転移学習による高精度AIアプリ開発**

> <https://www.udemy.com/share/102cllAEMecV5XQnkD/>
Djangoのバイク車判定アプリの作り方を教えてくれます。
これを応用して、花判定をするアプリにしました。

- **本：動かして学ぶ！ Python Django開発入門**

> <https://www.amazon.co.jp/gp/product/B07YY5RDPC/>
Djangoの基本を教えつつ、日記アプリを作る本です。
AWSへのデプロイ方法を主に参考にしました。