

Preferred Networks インターン選考2019 コーディング課題 Chainer（コンピュータービジョン）分野

機械学習を用いた画像処理の問題を扱います。課題2ではChainerの知識を前提とした問題となっています。

変更履歴

- 2019年4月19日：初版

回答にあたっての注意

- 課題1のソースコードの実装はOpenCV、ブラウザ等を自由にライブラリを使って構いません。
- 本問2のソースコードはchainer (<https://chainer.org/>)を用いて実装を行う事とします。
- 課題には自分だけで取り組んでください。この課題を他の応募者を含めた他人と共有・相談することを禁止します。課題期間中に GitHub の公開リポジトリ等にソースコードや問題をアップロードする行為も禁止します。漏洩の証拠が見つかった場合、その応募者は失格となります。ある応募者が別の応募者に回答をコピーさせた場合、双方の応募者が失格となります。
- 想定所要時間は最大2日です。全課題が解けていなくても提出は可能ですので、学業に支障の無い範囲で取り組んで下さい。

提出物

- 課題1、2のソースコードを、ビルド・実行方法および実行結果と合わせて提出してください。
- 課題1の実行結果のスクリーンショットや課題2の考察についてはpdfファイルにまとめて提出してください。

評価基準

提出物の評価にあたって以下のような要素を考慮します。

- ソースコードが他人が読んで理解しやすいものになっていること。
- ソースコードが正しいものであることを検証するために必要であればある程度の単体テストや検証のためのコードが書かれていること。
- 提出物を見て追試がしやすい形になっていること。
- レポートが要点についてわかりやすくまとまっていること。

一次選考後の面接で提出されたコードについて質問する可能性があります。

提出方法

上記の提出物を単一のパスワード無しzipファイルにまとめ、[こちらの専用フォーム](#)より応募してください。締切は日本時間2019年5月7日（火）12:00です。

問い合わせ

問題文に関して質問がある場合はintern2019-admin@preferred.jpまでご連絡ください。問題文に訂正が行われた場合は応募者全員にアナウンスいたします。なお、アプローチや解法に関する問い合わせにはお答えできません。

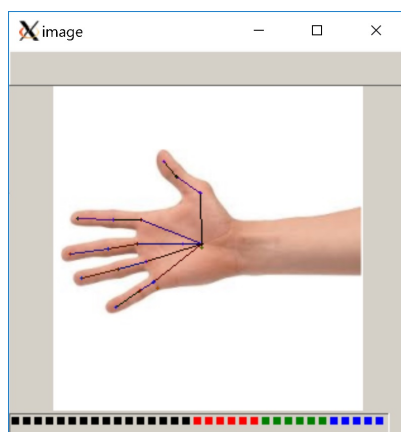
問題文

課題1: 手のポーズを推定するアノテーションツールを作成

本課題ではOpenCV等のライブラリを用いても、ブラウザ等を使っても構いません。GUIの問題では実行結果のスクリーンショットをまとめたpdfを提出してください。

ポーズ推定は深層学習の応用の一分野で、画像から人間のパーツの場所を推定します。[OpenPose](#)などが代表的です。

画像から機械学習を用いて手のポーズを推定するタスクを考えます。まず、手の画像に対してアノテーション（各画像に対して指先などのkey pointの正解データを設定する）を行い、データセットを簡単に構築するためのツールを作成します。



1. 手の画像（撮影してもインターネット上から取得しても構いません）を複数枚用意し、画像を順番にロードして表示する機能を作ってください。

'n'キーを押すと次の画像を読み込むようにしてください。

2. 画像をクリックした座標を取得して、点を表示する機能を作ってください。これが"key point"となります。
3. 's'キーを押すと点の位置をファイルに出力するようにしてください。フォーマットは自由です。
4. 一つの手に必要な点のセットを初期値として適当な位置にロードし、点と点をつなぐエッジも表示してください。
5. 各指を別々に扱うためにはクラス情報が必要です。それぞれの指を異なるクラスとして認識できるようツールを改善してください。
6. このようなシンプルなアノテータに対して、あると便利な機能を考察してください。実装も評価します。

課題2: CNNを用いて手のポーズを予測

本課題ではchainer (<https://chainer.org/>)を用いて実装を行う事とします。

1. 課題1で作成したアノテーションツールからデータをロードし、目標となるkey pointをheat mapに変換して入力画像と合わせて返すようなDataset Classを作ってください。

- heat mapは各key pointに対して1チャンネルの画像を割り当てて、[キーポイント数×縦×横]といった形式で返します。
- Dataset Classについては <https://docs.chainer.org/en/stable/reference/datasets.html> を参照してください。

2. VGG等のpretrained modelは画像の特徴量抽出器としてしばしば用いられます。VGGの中間層を特徴量としてCNNに入力し、heat mapをを予測するようなネットワークを作ってください。

- 入力解像度は224×224px、出力解像度は56×56pxとします。VGGのどの層の特徴量を使うかは自由です。
- VGG16:
<https://docs.chainer.org/en/stable/reference/generated/chainer.links.VGG16Layers.html>

3. heat mapはほとんど0の値を取るため、全て0を出力するような場合でもlossが下がってしまいます。

それを避けるための手法の一つとして、heat mapにweightをかけたlossを取る手法があります。そのようなloss関数を実装してください。

- 例: 0以上の値には $w=1$ 、0の値には $w=0.001$ のweightがかかるようなmean_absolute_error

4. key pointの学習をスムーズにする手法の一つとして、何段かに分けてrefineしていくというものがあります。

例えば、一段階目の予測結果:output1を出力し、画像特徴量とoutput1を更に2段階目のinputとしてoutput2を出力し〜、と繰り返して得られた出力のそれぞれに、lossをかけるというものです。

段階的にoutput1~3を出力するネットワークを作り、1で作ったデータセット、3で作った重み付きloss functionを用いて、実際にトレーニングを行ってください。

- なお、ここでは精度は問わないものとします。
- 学習済みモデルの提出は不要です。

5. トレーニングした予測結果をポイントに変換し、課題1のアノテータで読み込む機能を作ってください。

- heatmapを位置に変換する必要があります。
- 例: ガウシアンフィルタ等をかけてからチャンネルごとに最大の値を取る点の位置を調べる。

6. さらに精度を上げるための手法をいくつか検討し、考察してください。

- 単純にデータ数を増やすのは有効ですが、ここでは除くとします。
- 例えば、よりロバストな学習結果が得られるような方法を提案してください。