

Machine Learning Engineer Nanodegree

Proposta do projeto de conclusão

Leandro Humberto Vieira

Aplicação de machine learning e visão computacional na extração de informação sobre roupas femininas

Descrição do domínio

A imagem é a primeira impressão que uma pessoa tem ao ver alguém, é algo que nosso cérebro faz automaticamente para armazenar informações e é realizado a partir do momento em que começamos a interagir com esta pessoa. Uma imagem diferente do que uma pessoa é durante uma conversa causa um "ruído" na comunicação e uma confusão no nosso cérebro.

Hoje a imagem é uma via de comunicação pessoal tão importante quanto a fala, dessa forma surgiu a necessidade de profissionais que ensinam a arte da comunicação não-verbal, através de métodos e análises que permitem ensinar uma pessoa a se conhecer e, assim, expressar melhor seu estilo e personalidade através de sua imagem. O processo de consultoria de imagem (https://en.wikipedia.org/wiki/Image_consulting) ensina a pessoa a identificar seu estilo e sua personalidade, e após este passo, trabalhar com o objetivo de construir uma imagem pessoal mais positiva e coerente com ela mesma. A consultoria de imagem trabalha em sua grande parte com roupas, logo há algumas etapas do processo consultivo que visam limpar, organizar e classificar as roupas da cliente.

Por que a consultoria de imagem é importante

A imagem, o estilo e as vestimentas são fatores que regem ou influenciam vários aspectos da vida de uma pessoa, o que torna esta informação muito valiosa e digna de vários estudos como os descritos abaixo:

- Como a roupa afeta sua consciência (<http://www.dailymail.co.uk/sciencetech/article-2644076/You-DRESS-Clothing-significant-effect-self-esteem-confidence-claims-expert.html>) - Livro descrito no artigo (https://www.amazon.com/Mind-What-You-Wear-Psychology-ebook/dp/B00KBTB3NS/ref=sr_1_1?ie=UTF8&qid=1454635783&sr=8-1&keywords=Mind+What+You+Wear)
- Como a roupa impacta no seu sucesso profissional (<http://www.businessinsider.com/how-your-clothing-impacts-your-success-2014-8>)
- A correlação da vestimenta com seu relacionamento afetivo (<https://www.meetmindful.com/the-way-you-dress/>)
- Benefícios da organização das roupas (<https://www.janeisatomas.com.br/6-vantagens-de-ter-um-closet-organizado/>).

Motivação pessoal

Optei por este desafio pois minha esposa (<https://www.instagram.com/karinataniaconsultoria/?hl=pt-br>) é consultora de imagem há 1 ano e tem como especialidades o **atendimento online** e uma metodologia de como montar um armário inteligente (<https://www.youtube.com/watch?v=QSFPt3cG6CQ&t=2340s>). Desejo apoiá-la

neste assunto, utilizando tecnologia para dar vantagem competitiva ao produto dela, tornando o processo de transformação do armário mais interativo, automatizado e divertido.

Descrição do problema

O problema a ser resolvido se define em como ter uma visão estratégica do armário de uma mulher. Apenas olhando para o armário, por mais organizado que esteja, não é possível responder perguntas como:

- Quantas blusas pretas eu tenho?
- Quais são as peças essenciais que faltam no armário?
- Qual a proporção casual/trabalho que tenho?

Para responder tais perguntas, é necessário classificar as roupas entre os tipos básicos de vestimenta. De acordo com a Karina Tânia Consultoria, uma boa separação do armário consiste em agrupar as roupas de acordo com os 10 grupos abaixo:

1. Lingeries
2. Acessórios
3. Bolsas e sapatos
4. Blusas
5. Terceira peça
6. Calças
7. Shorts
8. Saias
9. Vestidos/Macacões/Macaquinhos
10. Roupas de Academia

Para classificar as roupas, deve-se observar em qual categoria ela pertence, através de sua forma básica. Para realizar esta classificação, a forma mais fácil de obter esta informação é através da imagem, devido a riqueza de detalhe que uma única peça de roupa pode conter.

Dados de entrada

Os dados de entrada foram pesquisados na internet e dois candidatos possuem um excelente encaixe na solução:

Fashion MNIST (<https://github.com/zalandoresearch/fashion-mnist>)



O Fashion MNIST é um conjunto de dados criado pela Zalando Research (<https://research.zalando.com/>), que contém 60 mil imagens de roupas no conjunto de treino e outras 10 mil no conjunto de teste. Este conjunto de dados tem como objetivo principal substituir o MNIST (<http://yann.lecun.com/exdb/mnist/>), pois o mesmo é um conjunto que não condiz com a realidade dos problemas de Deep Learning (<https://twitter.com/fchollet/status/852592598128615424>)

Apparel classification with Style
(<http://www.vision.ee.ethz.ch/~lbossard/projects/accv12/index.html>)



O Apparel Classification Set contém **80 mil imagens coloridas** retiradas da web através de crawlers e já classificadas em 15 grupos diferentes de categorias de roupa. Os dados deste conjunto são mais semelhantes com a realidade da aplicação final, pois mostram as roupas vestidas em pessoas em lugares naturais, porém contém uma série de ruídos nos dados, como fotos de caixas e fotos com zoom demais.

Talvez como solução final, o modelo seja treinado com uma subseleção dos dados do ACS, que contenham as fotos que mais se assemelham ao propósito da aplicação.

Descrição da solução

A solução envolve a construção de um aplicativo mobile que permitirá uma pessoa gerenciar seu armário com mais facilidade, através da construção de um "armário virtual". O processo consiste de um cadastro inicial, indicando a quantidade de peças que a pessoa possui no armário. Após isso o aplicativo iniciará como um armário vazio, e ele será preenchido organicamente pela cliente, através de fotos com o "look do dia"

(<https://www.instagram.com/p/BfoDmWrnYLb/?hl=pt-br&taken-by=karinataniaconsultoria>). Com as fotos, é o trabalho da solução classificar as roupas utilizadas nas fotos e adicioná-las ao armário virtual, com o devido cuidado de não criar a peça duplicadamente no armário.

Com a geração destes dados classificados, eles serão utilizá-los para mostrar estatísticas descritivas do armário para a cliente.

Modelo de comparação

Assim como descrito na [publicação do ACS](#)

(http://www.vision.ee.ethz.ch/~lbossard/projects/accv12/accv12_apparel-classification-with-style.pdf), o modelo a ser construído será comparado com abordagens simples dos algoritmos de classificação [random forest](#) (https://en.wikipedia.org/wiki/Random_forest) e [Support Vector Machines](#) (https://en.wikipedia.org/wiki/Support_vector_machine), para definir o baseline de performance a ser atingido pelo algoritmo de deep learning.

Por interesse pessoal, e devido à grande visibilidade da área recentemente, também incluírei como modelo de comparação, um classificador gerado através de [Auto Machine Learning](#) (<https://automl.info/>), utilizando a biblioteca [TPOT](#) (<http://epistasislab.github.io/tpot/>). Auto Machine Learning tem sido experimentado e observado por gigantes da indústria de tecnologia como [Google](#) (<https://cloud.google.com/automl/>) e [Microsoft](#) (<https://www.microsoft.com/en-us/research/blog/automl-challenge-leap-forward-machine-learning-competitions/>), então considero isto uma boa adição ao meu projeto.

Métrica de validação

A métrica de validação a ser utilizada pelo modelo será a [acurácia](#) (https://en.wikipedia.org/wiki/Accuracy_and_precision).

O processo de validação das métricas será dividido nas etapas abaixo:

- Criação de 3 modelos de machine learning de comparação: SVM, Random Forest e TPOT(AutoML)
- Criação de uma [rede neural convolucional](#) (https://pt.wikipedia.org/wiki/Rede_neural_convolucional).
- Treinamento e validação dos modelos
- Comparação da acurácia entre os modelos

Design do projeto

Juntando todas as peças mencionadas anteriormente, temos o workflow de projeto abaixo:

1. Entrada de dados

Os dados serão inseridos através de uma aplicação mobile, construída em [Flutter](#) (<https://flutter.io/>), que terá uma interface para upload de fotos da galeria do dispositivo. A aplicação será a mais simples possível, pois não é o foco deste projeto.

2. Transporte de dados

A foto que foi escolhida para a análise não será analisada no dispositivo, logo é necessário realizar o transporte dos dados para o servidor de análise. Este transporte será realizado através de uma requisição http.

3. Recepção dos dados

Para receber os dados através da requisição, será construído um simples servidor RESTful, através da biblioteca Flask (<http://flask.pocoo.org/>), o servidor irá fazer o papel de receber os arquivos e realizar as operações básicas de controle de sessão e chamada do algoritmo de classificação.

4. Pré processamento dos dados

Para realizar a classificação de um look, serão necessárias classificações distintas para cada peça de roupa que o compõe, para isso, considero utilizar como abordagem recortar a foto do look e extrair cada peça dele, para assim realizar a classificação utilizando apenas parte da imagem que contém a peça em questão.

5. Análise dos dados

Após a separação das peças de roupa em uma imagem, aplicar para cada uma separadamente a CNN de classificação, e retornar o resultado para a aplicação cliente.

6. Resultado final

Após a aplicação cliente receber o resultado na análise, será mostrado o recorte da peça para o usuário da aplicação, junto com o resultado da CNN, caso o resultado da CNN tenha um baixo grau de confiança, o usuário