

Aula 23 - Exemplo de Aplicação

João Florindo

Instituto de Matemática, Estatística e Computação Científica
Universidade Estadual de Campinas - Brasil
florindo@unicamp.br

Outline

- 1 Introdução
- 2 Janelas Deslizantes
- 3 Detecção de Texto
- 4 Segmentação
- 5 Síntese de Dados Artificiais
- 6 *Ceiling Analysis: Em Qual Parte do Pipeline se Concentrar?*

- OCR em texto escaneado já é um tópico praticamente resolvido.
- MAS, em imagens de cenários reais (*"in the wild"*), ainda é um grande desafio!



- OCR em texto escaneado já é um tópico praticamente resolvido.
- MAS, em imagens de cenários reais (*"in the wild"*), ainda é um grande desafio!



- OCR em texto escaneado já é um tópico praticamente resolvido.
- MAS, em imagens de cenários reais (*"in the wild"*), ainda é um grande desafio!



Pipeline Geral

1. Text detection



2. Character segmentation



3. Character classification

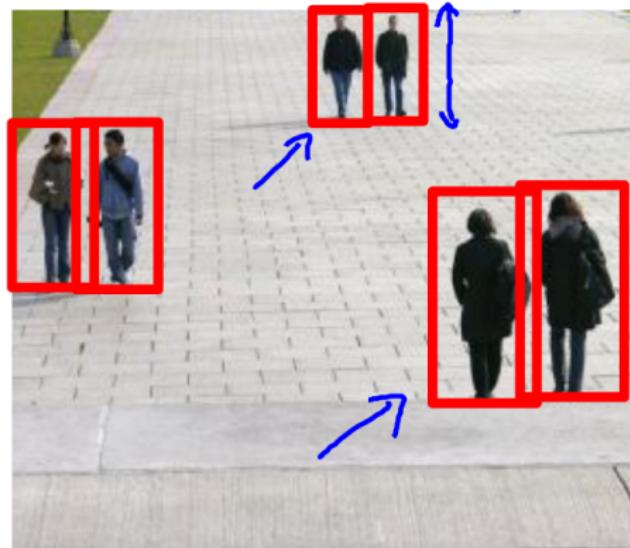


- Pode haver etapas mais avançadas, p.ex., análise de linguagem.
- Divisão em etapas permite que a equipe de trabalho possa ser dividida, ficando, p. ex., 1 a 5 engenheiros para cada etapa.

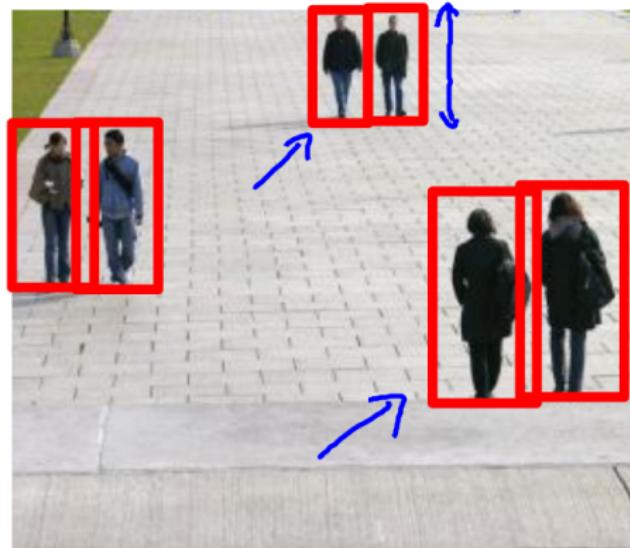
- Pode haver etapas mais avançadas, p.ex., análise de linguagem.
- Divisão em etapas permite que a equipe de trabalho possa ser dividida, ficando, p. ex., 1 a 5 engenheiros para cada etapa.

Outline

- 1 Introdução
- 2 Janelas Deslizantes
- 3 Detecção de Texto
- 4 Segmentação
- 5 Síntese de Dados Artificiais
- 6 *Ceiling Analysis: Em Qual Parte do Pipeline se Concentrar?*



- Tomamos janelas (*patches*) de treinamento x como sendo os pixels nesses patches.
- Se o *patch* contém um pedestre, ($y = 1$), senão, ($y = 0$).



- Tomamos janelas (*patches*) de treinamento x como sendo os pixels nesses patches.
- Se o *patch* contém um pedestre, ($y = 1$), senão, ($y = 0$).



- Em seguida, deslizamos uma janela do tamanho do *patch* por toda a imagem.
- Colocamos aquela região da imagem no conjunto de testes de uma regressão logística, por exemplo.



- Em seguida, deslizamos uma janela do tamanho do *patch* por toda a imagem.
- Colocamos aquela região da imagem no conjunto de testes de uma regressão logística, por exemplo.



- Em seguida, deslizamos uma janela do tamanho do *patch* por toda a imagem.
- Colocamos aquela região da imagem no conjunto de testes de uma regressão logística, por exemplo.



- A janela pode deslizar de 1 em 1 pixel ou com passo maior (*stride*) por eficiência computacional.
- Note que neste caso usualmente temos uma ideia do tamanho aproximado do *patch* que contém um pedestre.
- Mesmo assim, podemos repetir o algoritmo com um *patch* maior para detectar pedestres em outra escala.

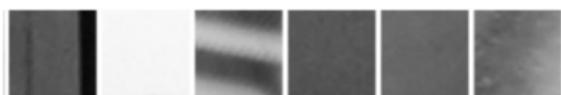
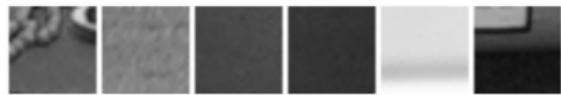
Outline

- 1 Introdução
- 2 Janelas Deslizantes
- 3 Detecção de Texto
- 4 Segmentação
- 5 Síntese de Dados Artificiais
- 6 *Ceiling Analysis: Em Qual Parte do Pipeline se Concentrar?*

PARTIDODNF

OPONICU

Positive examples ($y = 1$)



Negative examples ($y = 0$)



- ① Esquerda - Mapa de probabilidades $p(y = 1|x)$.
- ② Direita - Processamento: *Threshold* para tornar binário, expansão (pixels próximos a um branco setados para branco) e remoção de regiões sem formato de retângulo horizontal (esperado para um texto).

- Apenas um texto (verde) foi perdido, mas estava em um vidro transparente (difícil).

Outline

- 1 Introdução
- 2 Janelas Deslizantes
- 3 Detecção de Texto
- 4 Segmentação
- 5 Síntese de Dados Artificiais
- 6 *Ceiling Analysis: Em Qual Parte do Pipeline se Concentrar?*

- Mesma ideia de janela deslizante.

- ($y = 1$) quando dois caracteres aparecem cortados na mesma imagem.
- ($y = 0$) se a imagem contém apenas um único caractere.



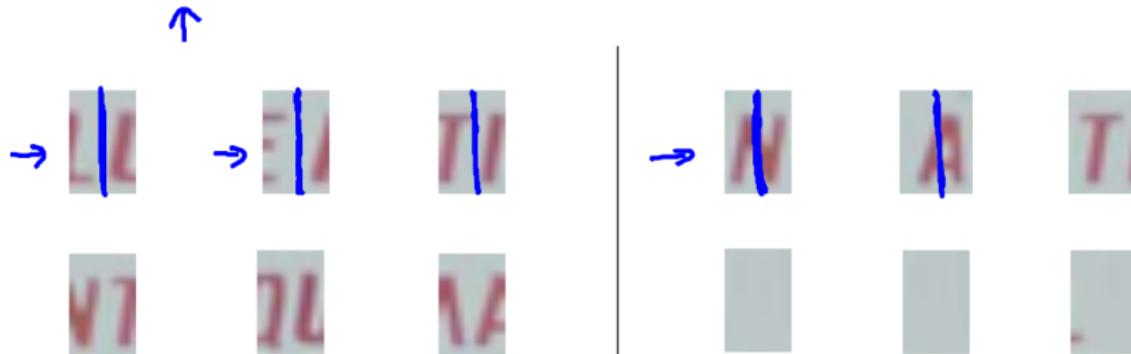
- Mesma ideia de janela deslizante.
- ($y = 1$) quando dois caracteres aparecem cortados na mesma imagem.
- ($y = 0$) se a imagem contém apenas um único caractere.



- Mesma ideia de janela deslizante.
- ($y = 1$) quando dois caracteres aparecem cortados na mesma imagem.
- ($y = 0$) se a imagem contém apenas um único caractere.



- Mesma ideia de janela deslizante.
- ($y = 1$) quando dois caracteres aparecem cortados na mesma imagem.
- ($y = 0$) se a imagem contém apenas um único caractere.



Positive examples ($y = 1$)

Negative examples ($y = 0$)

- Classificador vai reconhecer as regiões de separação entre caracteres.
- Cada caractere é classificado a partir de um conjunto de treinamento que contém imagens deste caractere.

- Classificador vai reconhecer as regiões de separação entre caracteres.
- Cada caractere é classificado a partir de um conjunto de treinamento que contém imagens deste caractere.

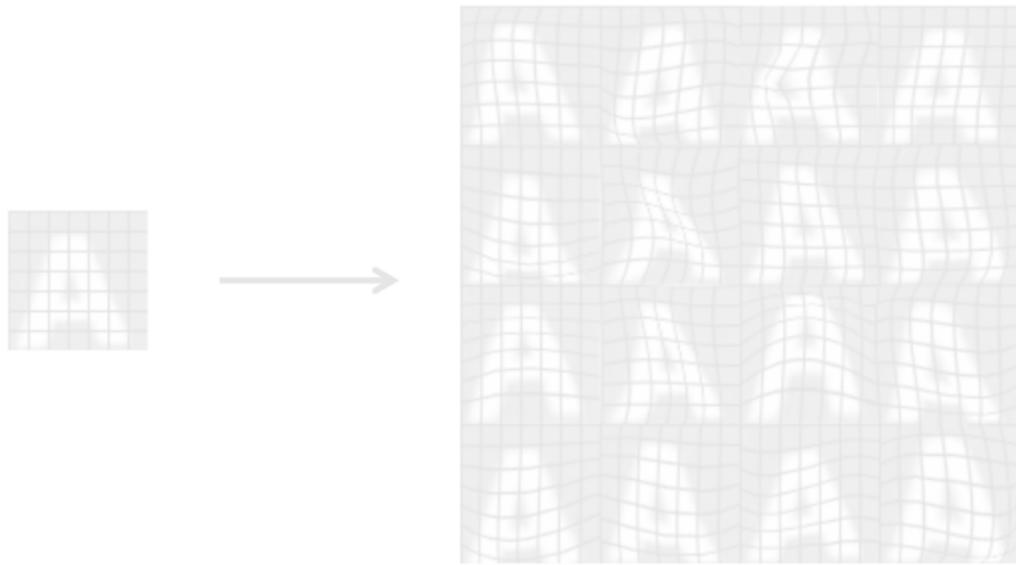
Outline

- 1 Introdução
- 2 Janelas Deslizantes
- 3 Detecção de Texto
- 4 Segmentação
- 5 Síntese de Dados Artificiais
- 6 *Ceiling Analysis: Em Qual Parte do Pipeline se Concentrar?*

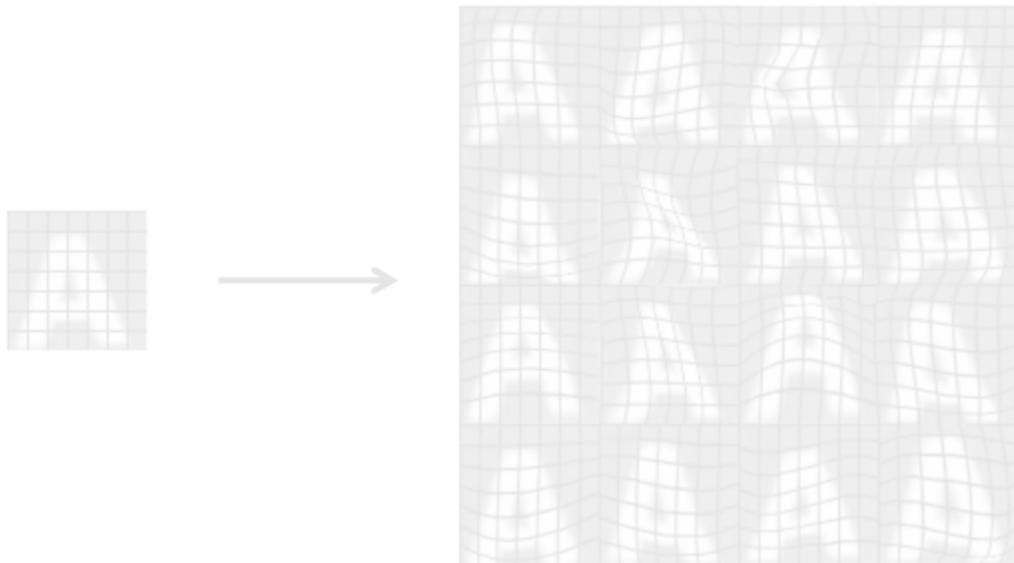
- Povoar treinamento com imagens do mundo real (esquerda) ou gerar dados artificiais (direita).



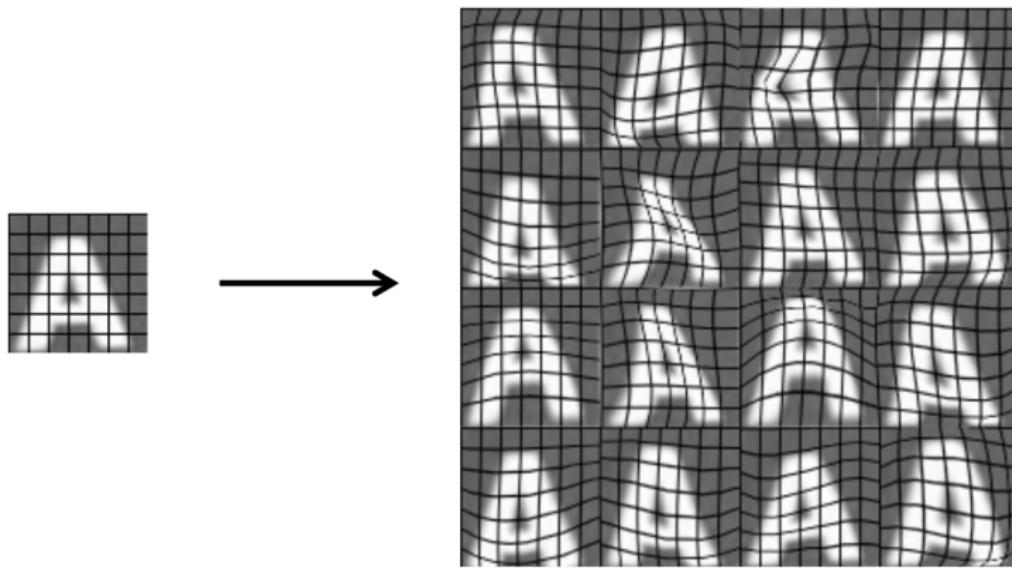
- Usar diferentes fontes e adicionar um fundo aleatório.
- Acrescentar transformações geométricas de espelhamento, rotação, escala, etc.
- Distorções sobre a imagem original:



- Usar diferentes fontes e adicionar um fundo aleatório.
- Acrescentar transformações geométricas de espelhamento, rotação, escala, etc.
- Distorções sobre a imagem original:



- Usar diferentes fontes e adicionar um fundo aleatório.
- Acrescentar transformações geométricas de espelhamento, rotação, escala, etc.
- Distorções sobre a imagem original:



- Estratégias desse tipo se aplicam também a outros tipos de dados.
- EX.: com áudio, simular uma conexão telefônica ruim ou um som de fundo qualquer.
- Distorções adicionadas devem ser representativas de distorções/ruídos no conjunto de teste.
- Adicionar ruído puramente aleatório ao dado não costuma ajudar.

PONTOS IMPORTANTES:

- ① Classificador deve ter um viés baixo antes de gastar tempo adquirindo mais dados (curvas de aprendizado). Se o viés for alto, acrescentar mais atributos, unidades na rede neural, etc.
- ② Quanto trabalho terá para obter 10 vezes mais dados do que tem atualmente?
 - Síntese de dados artificiais
 - Coletar e rotular os dados. Você mesmo vai fazer isso? Quanto tempo demora/custa cada exemplo? Quanto custaria no total? Vale a pena?
 - “Crowd source”: usuários fazendo trabalhos manuais pequenos por um pequeno valor. EX: Amazon Mechanical Turk. Problemas com a qualidade.

PONTOS IMPORTANTES:

- ① Classificador deve ter um viés baixo antes de gastar tempo adquirindo mais dados (curvas de aprendizado). Se o viés for alto, acrescentar mais atributos, unidades na rede neural, etc.
- ② Quanto trabalho terá para obter 10 vezes mais dados do que tem atualmente?
 - Síntese de dados artificiais
 - Coletar e rotular os dados. Você mesmo vai fazer isso? Quanto tempo demora/custa cada exemplo? Quanto custaria no total? Vale a pena?
 - “*Crowd source*”: usuários fazendo trabalhos manuais pequenos por um pequeno valor. EX.: Amazon Mechanical Turk. Problemas com a qualidade.

PONTOS IMPORTANTES:

- ① Classificador deve ter um viés baixo antes de gastar tempo adquirindo mais dados (curvas de aprendizado). Se o viés for alto, acrescentar mais atributos, unidades na rede neural, etc.
- ② Quanto trabalho terá para obter 10 vezes mais dados do que tem atualmente?
 - Síntese de dados artificiais
 - Coletar e rotular os dados. Você mesmo vai fazer isso? Quanto tempo demora/custa cada exemplo? Quanto custaria no total? Vale a pena?
 - “*Crowd source*”: usuários fazendo trabalhos manuais pequenos por um pequeno valor. EX.: Amazon Mechanical Turk. Problemas com a qualidade.

PONTOS IMPORTANTES:

- ① Classificador deve ter um viés baixo antes de gastar tempo adquirindo mais dados (curvas de aprendizado). Se o viés for alto, acrescentar mais atributos, unidades na rede neural, etc.
- ② Quanto trabalho terá para obter 10 vezes mais dados do que tem atualmente?
 - Síntese de dados artificiais
 - Coletar e rotular os dados. Você mesmo vai fazer isso? Quanto tempo demora/custa cada exemplo? Quanto custaria no total? Vale a pena?
 - “*Crowd source*”: usuários fazendo trabalhos manuais pequenos por um pequeno valor. EX.: Amazon Mechanical Turk. Problemas com a qualidade.

PONTOS IMPORTANTES:

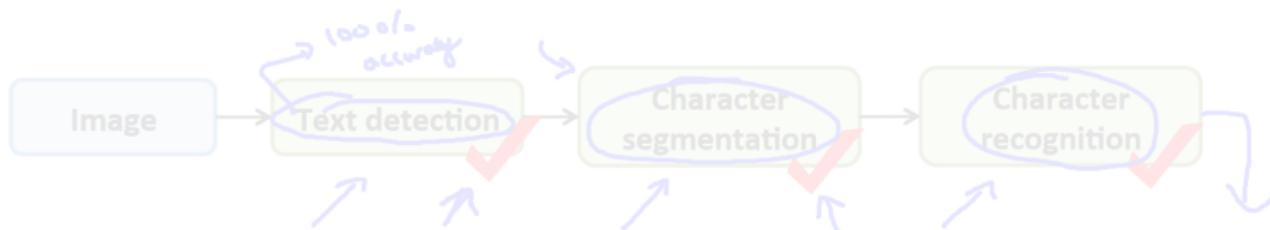
- ① Classificador deve ter um viés baixo antes de gastar tempo adquirindo mais dados (curvas de aprendizado). Se o viés for alto, acrescentar mais atributos, unidades na rede neural, etc.
- ② Quanto trabalho terá para obter 10 vezes mais dados do que tem atualmente?
 - Síntese de dados artificiais
 - Coletar e rotular os dados. Você mesmo vai fazer isso? Quanto tempo demora/custa cada exemplo? Quanto custaria no total? Vale a pena?
 - “*Crowd source*”: usuários fazendo trabalhos manuais pequenos por um pequeno valor. EX.: Amazon Mechanical Turk. Problemas com a qualidade.

Outline

- 1 Introdução
- 2 Janelas Deslizantes
- 3 Detecção de Texto
- 4 Segmentação
- 5 Síntese de Dados Artificiais
- 6 *Ceiling Analysis: Em Qual Parte do Pipeline se Concentrar?*

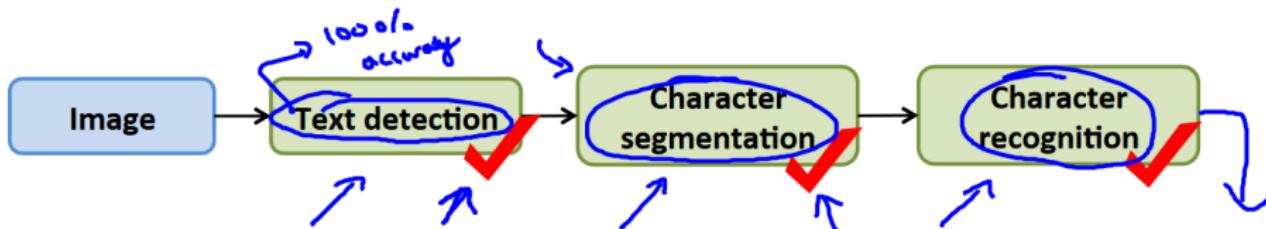
Ceiling analysis (limitante superior de desempenho)

Manualmente fazer com que uma etapa do *pipeline* tenha 100% de acurácia e verificar nessas circunstâncias então qual a acurácia na etapa seguinte.



Ceiling analysis (limitante superior de desempenho)

Manualmente fazer com que uma etapa do *pipeline* tenha 100% de acurácia e verificar nessas circunstâncias então qual a acurácia na etapa seguinte.



Exemplo

- Acurácia do sistema original: 72%.
- Forçamos 100% de acurácia na detecção de texto, detectando manualmente.
- Acurácia do sistema como um todo sobe para 89%.
- Segmentação manual para obter 100% de acurácia e acurácia geral vai para 90%.
- Finalmente, reconhecimento manual de caractere e, como se espera, acurácia geral chega em 100%.

Exemplo

- Acurácia do sistema original: 72%.
- Forçamos 100% de acurácia na detecção de texto, detectando manualmente.
- Acurácia do sistema como um todo sobe para 89%.
- Segmentação manual para obter 100% de acurácia e acurácia geral vai para 90%.
- Finalmente, reconhecimento manual de caractere e, como se espera, acurácia geral chega em 100%.

Exemplo

- Acurácia do sistema original: 72%.
- Forçamos 100% de acurácia na detecção de texto, detectando manualmente.
- Acurácia do sistema como um todo sobe para 89%.
- Segmentação manual para obter 100% de acurácia e acurácia geral vai para 90%.
- Finalmente, reconhecimento manual de caractere e, como se espera, acurácia geral chega em 100%.

Exemplo

- Acurácia do sistema original: 72%.
- Forçamos 100% de acurácia na detecção de texto, detectando manualmente.
- Acurácia do sistema como um todo sobe para 89%.
- Segmentação manual para obter 100% de acurácia e acurácia geral vai para 90%.
- Finalmente, reconhecimento manual de caractere e, como se espera, acurácia geral chega em 100%.

Exemplo

- Acurácia do sistema original: 72%.
- Forçamos 100% de acurácia na detecção de texto, detectando manualmente.
- Acurácia do sistema como um todo sobe para 89%.
- Segmentação manual para obter 100% de acurácia e acurácia geral vai para 90%.
- Finalmente, reconhecimento manual de caractere e, como se espera, acurácia geral chega em 100%.

Exemplo

Sistema geral	72%
Detecção de texto	89%
Segmentação de caractere	90%
Reconhecimento de caractere	100%

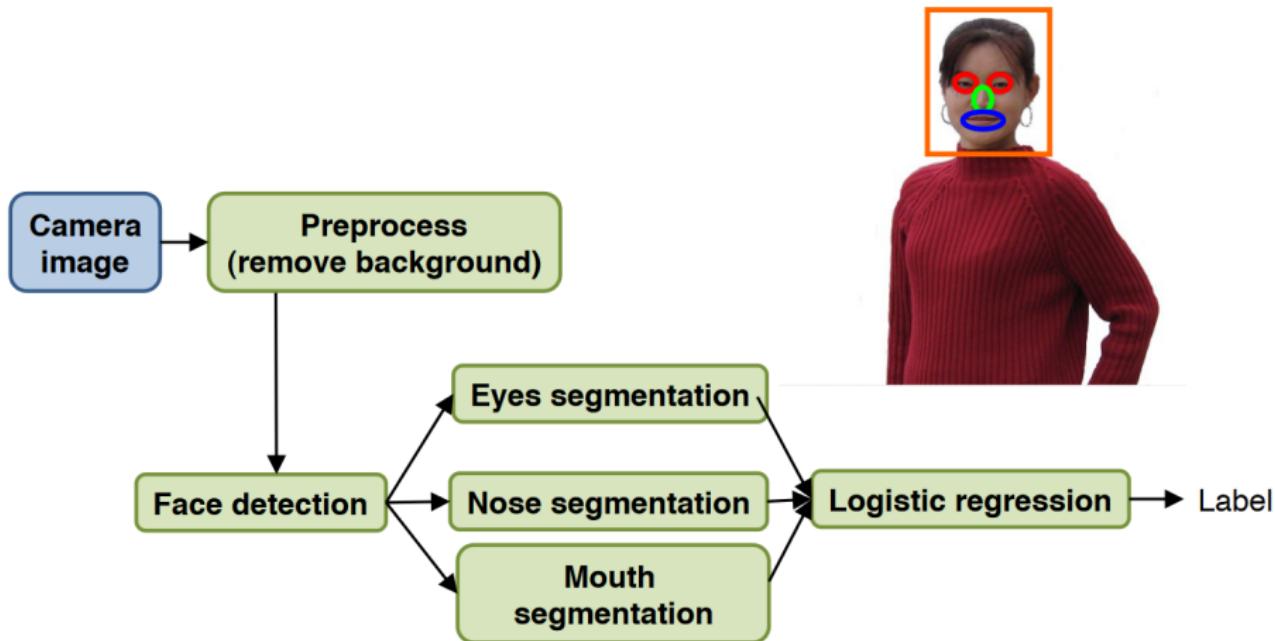
- Com a detecção de texto, acurácia aumentou 17%. Com a segmentação 1% e com o reconhecimento 10%.
- Então devemos, em primeiro lugar, concentrar nosso tempo e esforços no sistema de detecção de texto, em seguida no reconhecimento de caractere.

Exemplo

Sistema geral	72%
Detecção de texto	89%
Segmentação de caractere	90%
Reconhecimento de caractere	100%

- Com a detecção de texto, acurácia aumentou 17%. Com a segmentação 1% e com o reconhecimento 10%.
- Então devemos, em primeiro lugar, concentrar nosso tempo e esforços no sistema de detecção de texto, em seguida no reconhecimento de caractere.

Outro Exemplo



Outro Exemplo

- Removemos o fundo manualmente com 100% de acurácia.
- Detectamos a face, segmentamos manualmente os olhos (usualmente a parte mais importante em reconhecimento facial), o nariz e a boca.
- Finalmente classificamos manualmente tendo certeza de 100% de precisão.
- Após cada etapa, reavaliarmos a acurácia.

Outro Exemplo

- Removemos o fundo manualmente com 100% de acurácia.
- Detectamos a face, segmentamos manualmente os olhos (usualmente a parte mais importante em reconhecimento facial), o nariz e a boca.
- Finalmente classificamos manualmente tendo certeza de 100% de precisão.
- Após cada etapa, reavaliarmos a acurácia.

Outro Exemplo

- Removemos o fundo manualmente com 100% de acurácia.
- Detectamos a face, segmentamos manualmente os olhos (usualmente a parte mais importante em reconhecimento facial), o nariz e a boca.
- Finalmente classificamos manualmente tendo certeza de 100% de precisão.
- Após cada etapa, reavaliarmos a acurácia.

Outro Exemplo

- Removemos o fundo manualmente com 100% de acurácia.
- Detectamos a face, segmentamos manualmente os olhos (usualmente a parte mais importante em reconhecimento facial), o nariz e a boca.
- Finalmente classificamos manualmente tendo certeza de 100% de precisão.
- Após cada etapa, reavaliarmos a acurácia.

Outro Exemplo

Sistema geral	85%
Pré-processamento (remoção de fundo)	85.1%
Detecção de face	91%
Detecção dos olhos	95%
Detecção do nariz	96%
Detecção da boca	97%
Classificação	100%

- Remoção de fundo trouxe um ganho de apenas 0.1%, detecção de face nos deu 5.9%, detecção dos olhos 4%, detecção do nariz 1%, detecção da boca 1% e classificação nos trouxe 3%.
- Concentrar-se em melhorar, pela ordem, os algoritmos de detecção de face, detecção de olhos e o classificador.
- Não se deve confiar na intuição (*gut feeling*) para a tomada de decisões.
- Casos de equipes que gastaram um ano e meio em algoritmos de remoção de fundo para reconhecimento facial e descobriram que o ganho era marginal.

- Remoção de fundo trouxe um ganho de apenas 0.1%, detecção de face nos deu 5.9%, detecção dos olhos 4%, detecção do nariz 1%, detecção da boca 1% e classificação nos trouxe 3%.
- Concentrar-se em melhorar, pela ordem, os algoritmos de detecção de face, detecção de olhos e o classificador.
- Não se deve confiar na intuição (*gut feeling*) para a tomada de decisões.
- Casos de equipes que gastaram um ano e meio em algoritmos de remoção de fundo para reconhecimento facial e descobriram que o ganho era marginal.

- Remoção de fundo trouxe um ganho de apenas 0.1%, detecção de face nos deu 5.9%, detecção dos olhos 4%, detecção do nariz 1%, detecção da boca 1% e classificação nos trouxe 3%.
- Concentrar-se em melhorar, pela ordem, os algoritmos de detecção de face, detecção de olhos e o classificador.
- Não se deve confiar na intuição (*gut feeling*) para a tomada de decisões.
- Casos de equipes que gastaram um ano e meio em algoritmos de remoção de fundo para reconhecimento facial e descobriram que o ganho era marginal.

- Remoção de fundo trouxe um ganho de apenas 0.1%, detecção de face nos deu 5.9%, detecção dos olhos 4%, detecção do nariz 1%, detecção da boca 1% e classificação nos trouxe 3%.
- Concentrar-se em melhorar, pela ordem, os algoritmos de detecção de face, detecção de olhos e o classificador.
- Não se deve confiar na intuição (*gut feeling*) para a tomada de decisões.
- Casos de equipes que gastaram um ano e meio em algoritmos de remoção de fundo para reconhecimento facial e descobriram que o ganho era marginal.