



## **CORREÇÃO AUTOMÁTICA DE AVALIAÇÕES OBJETIVAS COM LEITURA DO CARTÃO-RESPOSTA POR VISÃO COMPUTACIONAL**

Autores: Lucas Ceroni MENONCIN, Ricardo ANTONELLO.

Identificação autores: Projeto de pesquisa do Edital 79-2016; IFC-Campus Luzerna.

### **RESUMO**

Visando facilitar e agilizar a correção de provas objetivas, apresentamos aqui um algoritmo escrito em Python utilizando a biblioteca OpenCV (Open Source Computer Vision) que faz a leitura da imagem e a identificação das respostas. O algoritmo recebe um streaming de vídeo capturado pela webcam e faz a análise frame a frame. Ao identificar um retângulo com 50 retângulos menores no frame (quantidade de alternativas), o algoritmo alinha a imagem e faz a identificação das respostas assinaladas no gabarito. O conjunto de respostas é comparado com o gabarito para que seja calculada a nota final da avaliação.

### **INTRODUÇÃO E JUSTIFICATIVA**

O projeto em questão surgiu da necessidade de agilizar processos de correção de avaliações objetivas, visando poupar tempo e energia de professores ocupados. O algoritmo apresentado tem como objetivo através da visão computacional reconhecer o cartão-resposta de uma avaliação objetiva e retirar as respostas assinaladas, para que assim possa ser comparado com um gabarito correto pré-determinado e seja realizada a correção automática.

O programa desenvolvido trabalha com duas entradas, o gabarito correto da prova previamente inserido, e uma entrada de vídeo em tempo real, podendo ser de uma webcam comum. Após reconhecidas as alternativas marcadas uma imagem é salva e a saída então é um vetor numérico de 10 posições para cada uma das alternativas, com suas posições "0" a "4" para representar de "A" a "E", e também a nota da avaliação.

Para ilustrar melhor o reconhecimento foi inserido ao algoritmo um comportamento especial: Após o reconhecimento e a geração de um vetor



respostas, os contornos das alternativas marcadas na imagem salva recebem a cor verde se estiver correta, e vermelho caso contrário.

## METODOLOGIA

Foram diversas tentativas para alcançar um algoritmo conseguisse abranger uma variação de luminosidade, ângulo e tamanho de imagem e fizesse a leitura correta da mesma. A procura de padrões é muito complicada trabalhando com imagens de fontes distintas. O algoritmo em que chegamos utiliza como base os contornos fazendo uso de algumas bibliotecas específicas.

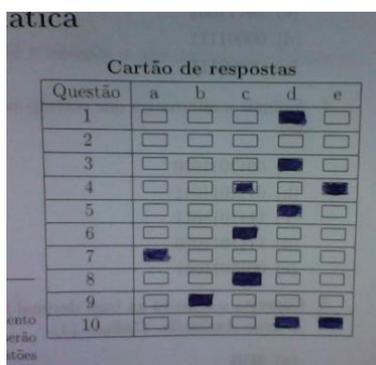
Para o desenvolvimento do algoritmo foi utilizada a linguagem de programação Python, e bibliotecas como OpenCV - Open Source Computer Vision Library (OPENCV, 2017), Numpy - Numeric Python (NUMPY, 2017) e imutils (ROSEBROCK, 2017), uma biblioteca auxiliar usada aqui para manipular (nesse caso) principalmente contornos da imagem.

A biblioteca OpenCV possui mais de 2.500 algoritmos otimizados, podendo ser usados para detectar e reconhecer faces, identificar objetos, classificar movimentos humanos em tempo real, seguir movimentos em câmeras, seguir objetos, dentre outras inúmeras utilidades (OPENCV, 2017).

## RESULTADOS E DISCUSSÕES

Para testar o algoritmo, inserimos D para todas as respostas corretas, e então, na entrada de vídeo mostramos o gabarito da prova, no momento em que ele reconhece o gabarito passa a trabalhar com as seguintes imagens.

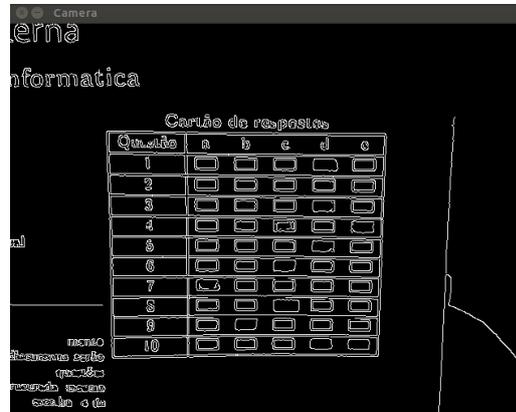
**Imagem 1** – Imagem real capturada pela câmera.



Fonte: Os autores.



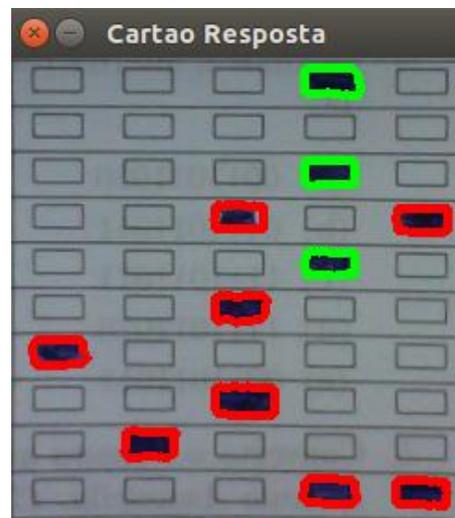
**Imagem 2** – Imagem com um filtro de contornos.



Fonte: Os autores.

O algoritmo então recorta apenas o gabarito com todas as 50 alternativas e compara com as respostas corretas:

**Imagem 3** – Gabarito recortado e respostas já marcadas.



Fonte: Os autores.

**Imagem 4** – Terminal.

```
Gabarito: [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3]
Respostas: [3, -1, 3, -1, 3, 2, 0, 2, 1, -1]
Nota: 3.0
```

Fonte: Os autores.



Note que na imagem 4, para alternativas é melhor trabalhar com números em nosso caso, em que “0” corresponde a “A”, e sucessivamente até “4” ser equivalente a “E”. Temos então o gabarito sendo todas as corretas como “D”. Já nas respostas notamos além das alternativas de “0” a “4”, o número “-1”, sendo ele o indicador de erro, para os casos de nenhuma, ou mais de uma alternativa serem marcadas na mesma questão.

### CONSIDERAÇÕES FINAIS

O reconhecimento do cartão resposta em condições ideais (imagem perfeita) foi extremamente positivo, porém, o que buscamos é que possa ser usado por qualquer pessoa, o que implica em imagens com tamanho, ângulo e luminosidade diferentes, isso acaba se tornando um grande obstáculo no decorrer da programação.

Mesmo tendo que lidar com os retângulos pintados de forma imperfeita (extrapolando seus limites) e com as barreiras das variáveis de imagens de câmeras completamente distintas, o algoritmo tem se mostrado funcional, conseguindo reconhecer com precisão a maioria dos casos, e sendo aprimorado cada dia mais.

### REFERÊNCIAS

NUMPY. **NumPy**. Disponível em: < <http://www.numpy.org/>>. Acesso em 27 de jul. 2017

OPENCV. **About OpenCV**. Disponível em: <<http://opencv.org/about.html>>. Acesso em: 11 de jul. 2017.

PYTHON. **Python is a programming language that lets you work quickly and integrate systems more effectively**. Disponível em: <<https://www.python.org/>>. Acesso em: 11 de jul. 2017.





**X MICTI**  
Campus Camboriú

Ministério Nacional de Inovação  
Ciência e Tecnologia Interdisciplinar

8 E 9 DE NOVEMBRO

**III IF CULTURA**

ROSEBROCK, Adrian. **I just open sourced my personal imutils package: A series of OpenCV convenience functions.** Disponível em: <<http://www.pyimagesearch.com/2015/02/02/just-open-sourced-personal-imutils-package-series-opencv-convenience-functions/>>. Acesso em 11 de jul. 2017.



**INSTITUTO FEDERAL**  
Catarinense