



Disciplina Introdução à Computação Visual	Turmas
Professor William Robson Schwartz	

**Entrega: 19/09/2017 até às 23h55 (via moodle)**

## Trabalho Prático 1: Reconhecimento de Cartão-Resposta

O objetivo deste trabalho é reconhecer o preenchimento de um cartão-resposta, como o mostrado na Figura 1. *O trabalho deve ser feito individualmente ou em duplas.*

1	a	b	c	d	e
2	a	b	c	d	e
3	a	b	c	d	e
4	a	b	c	d	e
5	a	b	c	d	e
6	a	b	c	d	e
7	a	b	c	d	e
8	a	b	c	d	e
9	a	b	c	d	e
10	a	b	c	d	e
11	a	b	c	d	e
12	a	b	c	d	e
13	a	b	c	d	e
14	a	b	c	d	e
15	a	b	c	d	e
16	a	b	c	d	e
17	a	b	c	d	e
18	a	b	c	d	e
19	a	b	c	d	e
20	a	b	c	d	e
21	a	b	c	d	e
22	a	b	c	d	e
23	a	b	c	d	e
24	a	b	c	d	e
25	a	b	c	d	e
26	a	b	c	d	e
27	a	b	c	d	e
28	a	b	c	d	e
29	a	b	c	d	e
30	a	b	c	d	e

Figura 1: Exemplo de cartão-resposta (não preenchido).

## O que deve ser feito

Implementar um método para efetuar o reconhecimento de cartão-resposta. A entrada será um cartão preenchido e digitalizado, como mostrado na Figura 2. O reconhecedor deve ser feito utilizando linguagem Python, com a utilização da biblioteca OpenCV versão 3, dentro do ambiente *Jupyter Note-*

*book*<sup>1</sup>. Também podem ser utilizadas bibliotecas já instaladas, mas não podem ser utilizadas bibliotecas específicas que requerem instalação adicional.

O método deve ser capaz de reconhecer a marcação efetuada para cada uma das trinta respostas contidas no cartão. Como saída, para cada questão, deve ser impresso na tela a letra correspondente a alternativa marcada, a palavra “Branco” caso nenhuma letra tenha sido marcada para aquela questão (ou o preenchimento das caixas esteja abaixo de um limiar definido no programa, como exemplo a questão 5 da Figura 2), ou a palavra “Nulo” caso mais de uma alternativa tenha sido marcada (como por exemplo as questões 8 e 23 da Figura 2). A Figura 3 mostra a saída esperada quando se utiliza a imagem da Figura 2 como entrada.

Questão	a	b	c	d	e
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					
28					
29					
30					

0002

Figura 2: Exemplo de cartão-resposta preenchido e digitalizado (imagem de entrada para o algoritmo).

<sup>1</sup><http://jupyter.org/>

1: A	16: A
2: Branco	17: B
3: A	18: Branco
4: C	19: B
5: Branco	20: D
6: C	21: C
7: C	22: C
8: Nulo	23: Nulo
9: Branco	24: C
10: C	25: D
11: B	26: A
12: D	27: C
13: A	28: D
14: C	29: C
15: D	30: A

Figura 3: Exemplo do arquivo texto de saída para o programa a ser desenvolvido (considerando a imagem da Figura 2 como entrada).

Conforme ilustrado na Figura 4, o cartão de respostas pode estar rotacionado e escalado dentro da imagem. Portanto, o seu algoritmo deve tratar casos dessa natureza.

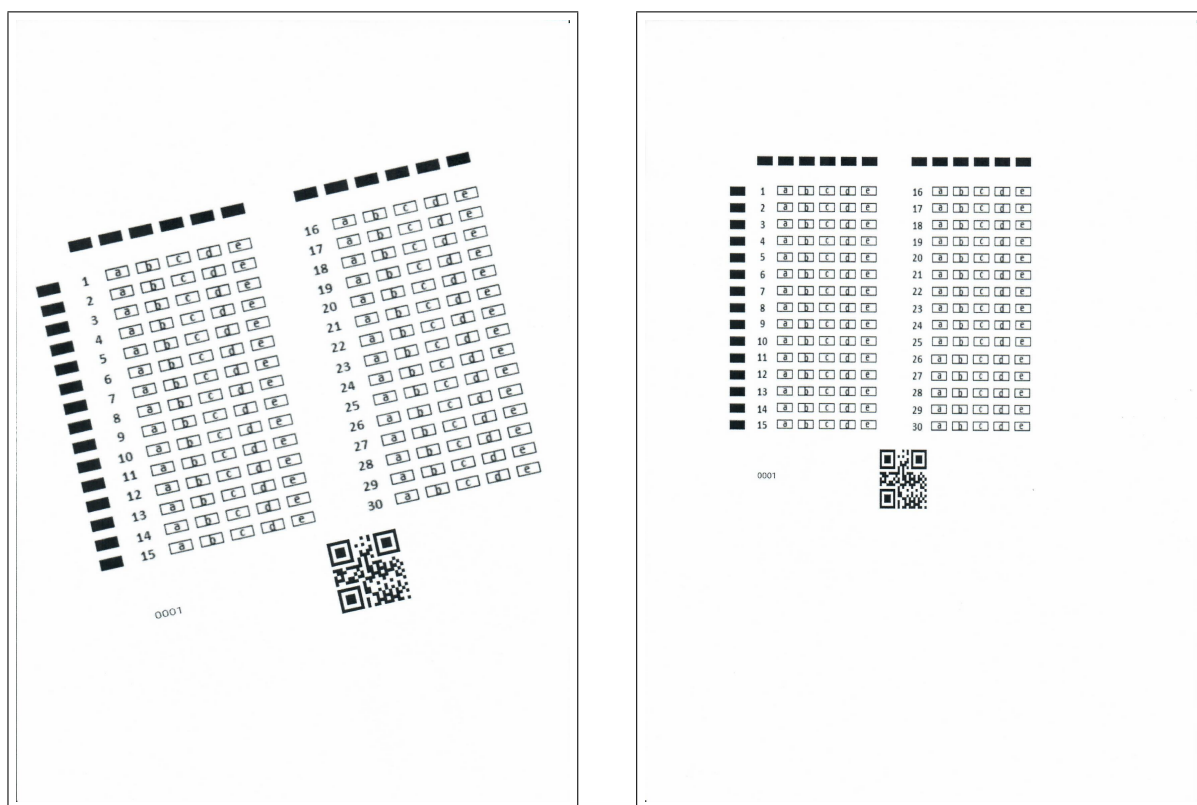


Figura 4: Exemplo de cartão-resposta preenchido e digitalizado (imagem de entrada para o algoritmo). Note que o cartão de respostas pode estar rotacionado e escalado dentro da imagem.

Modularize o seu código criando uma (ou mais) célula do tipo *Code* para cada etapa do método. Por exemplo, crie uma célula para efetuar a limiarização da imagem, outra para localizar as caixas, outra para determinar quais caixas estão preenchidas, assim por diante. Ao final de cada etapa, se pertinente, apresente resultados intermediários (isso facilita na correção e na compreensão do método). Também crie células do tipo *Markdown* para explicar os passos e as escolhas feitas para o método.

## O que deve ser entregue

O notebook salvo no ambiente Jupyter deve ser submetido no Moodle.

## Execução

O notebook submetido será carregado no Jupyter e outra imagem será carregada e o código será executado passo a passo para efetuar a correção. Portanto, é importante que as etapas do método sejam implementadas em células com visualização dos resultados intermediários, juntamente com as devidas explicações em comentários e em células do tipo *Markdown*.