



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA - CI
DEPARTAMENTO DE COMPUTAÇÃO CIENTÍFICA

Projeto 1

INTRODUÇÃO AO PROCESSAMENTO DIGITAL DE IMAGENS **LEONARDO VIDAL BATISTA**

ENILSON LIMA DA COSTA JUNIOR

João Pessoa - PB
Agosto de 2025

Introdução

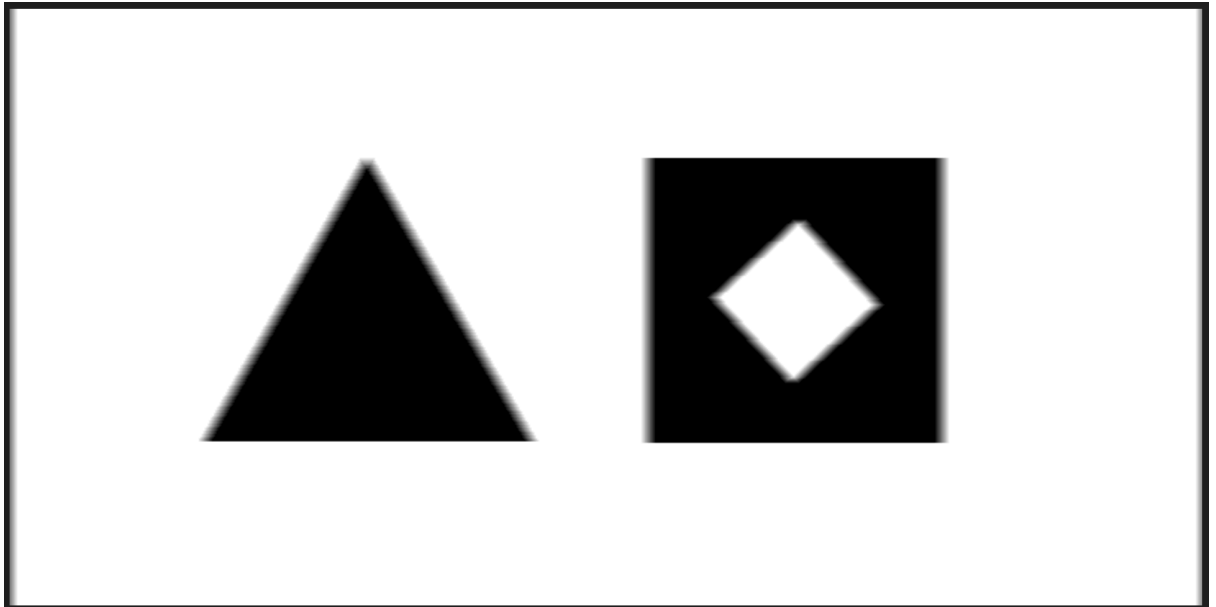
O processamento digital de imagens (PDI) é uma área da ciência da computação voltada para análise e transformação de imagens por meio de algoritmos. Entre as técnicas fundamentais estão a convolução e a correlação, que utilizam máscaras (kernels) para realizar operações como detecção de bordas, suavização e realce. O objetivo deste trabalho é implementar em linguagem Go um sistema que aplique diferentes kernels a imagens digitais, analisando seus efeitos e resultados.

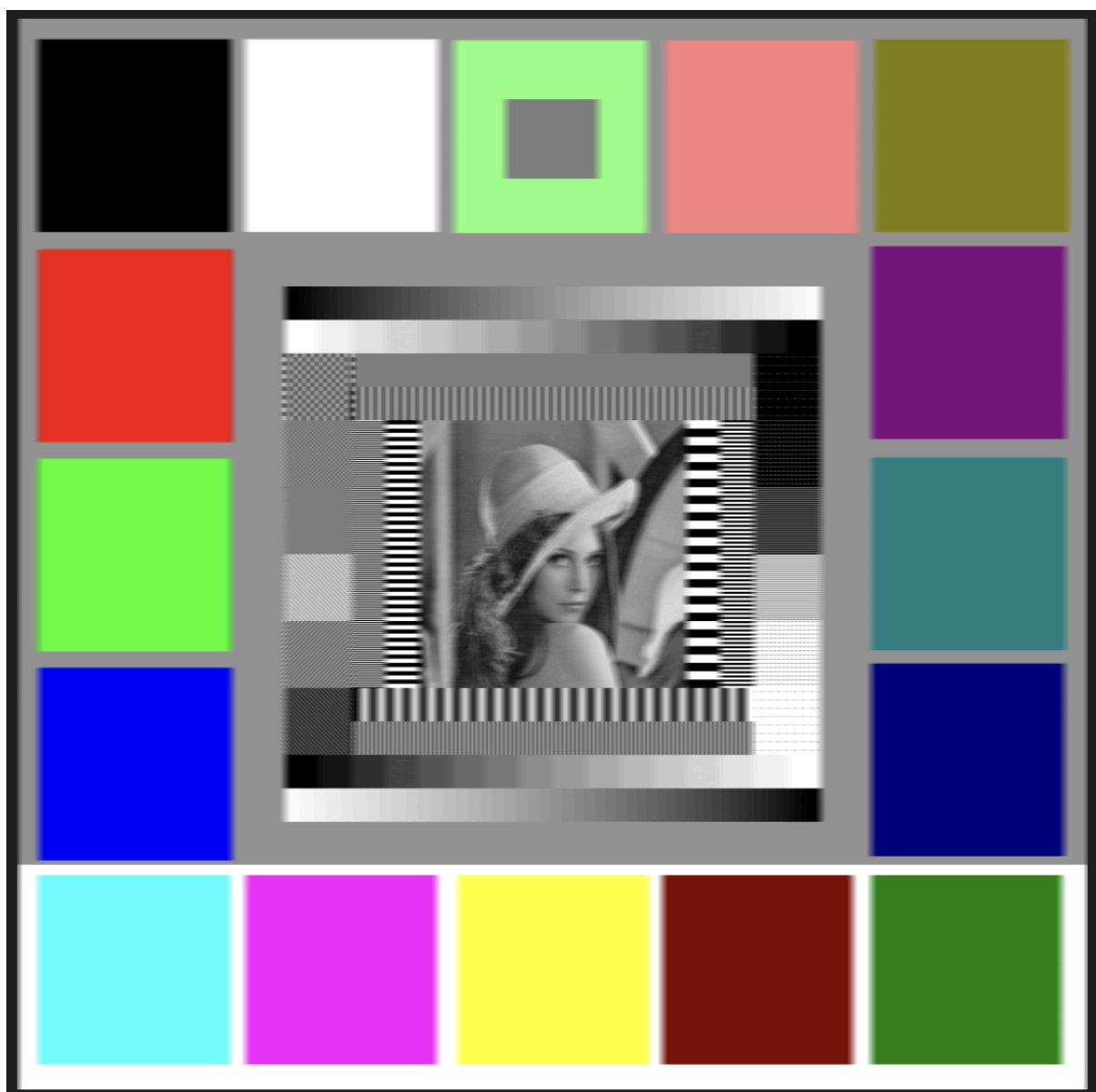
Materiais e Métodos

O trabalho foi desenvolvido em linguagem Go, utilizando as bibliotecas padrão para manipulação de imagens. Foram implementados módulos para leitura, conversão, aplicação de kernels e salvamento dos resultados. As máscaras utilizadas incluem filtros de suavização (Box e Gaussiano) e de detecção de bordas (Sobel). O método consistiu em aplicar cada kernel a imagens de teste e comparar os efeitos visuais obtidos.

Resultados

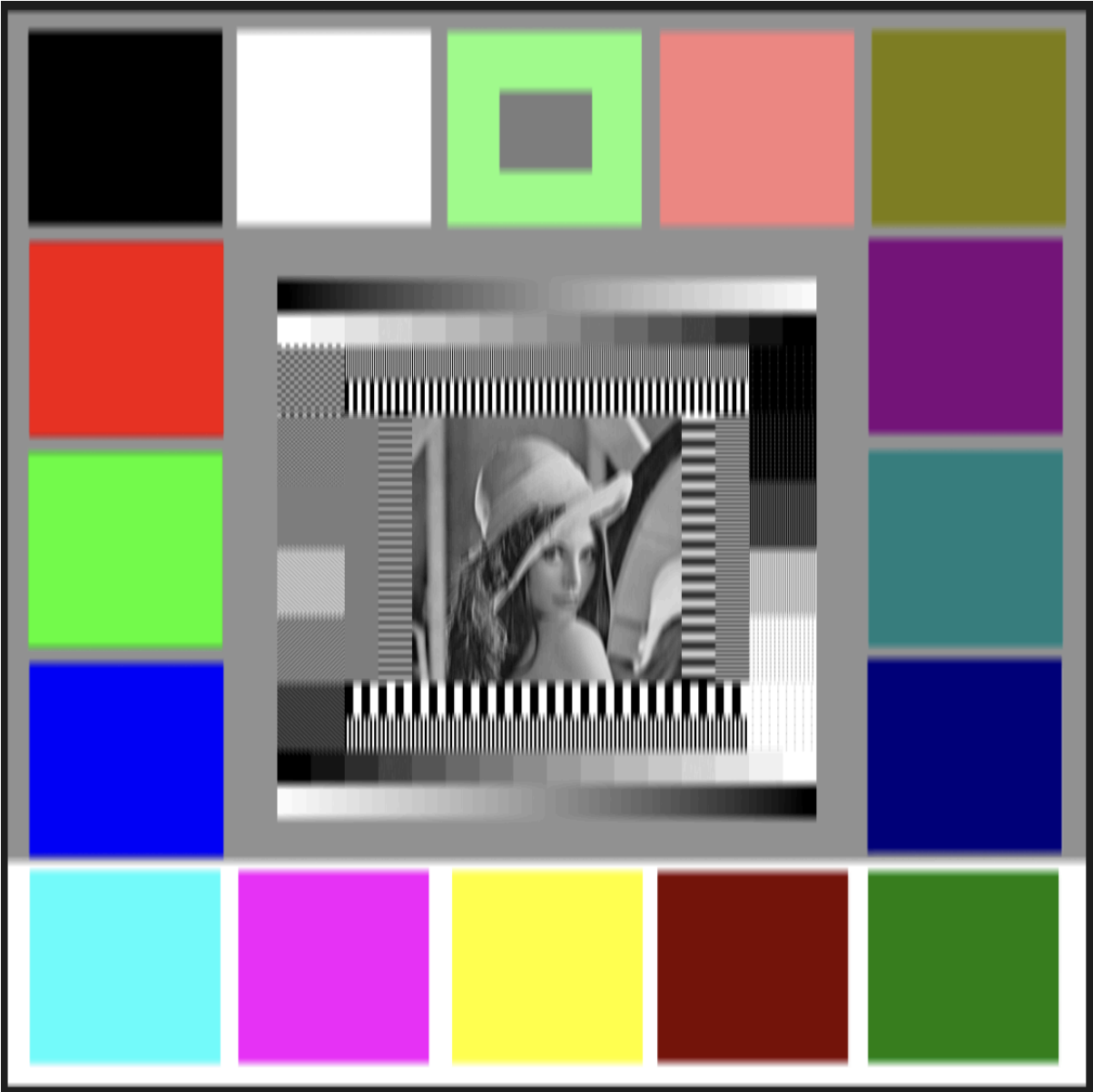
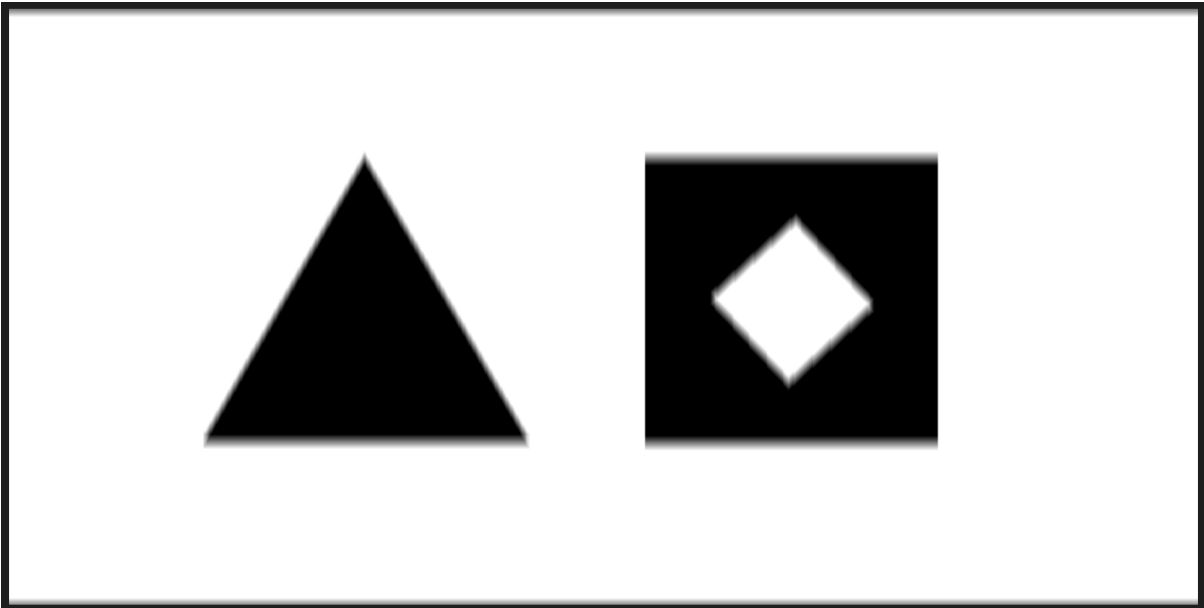
- Box1x10:





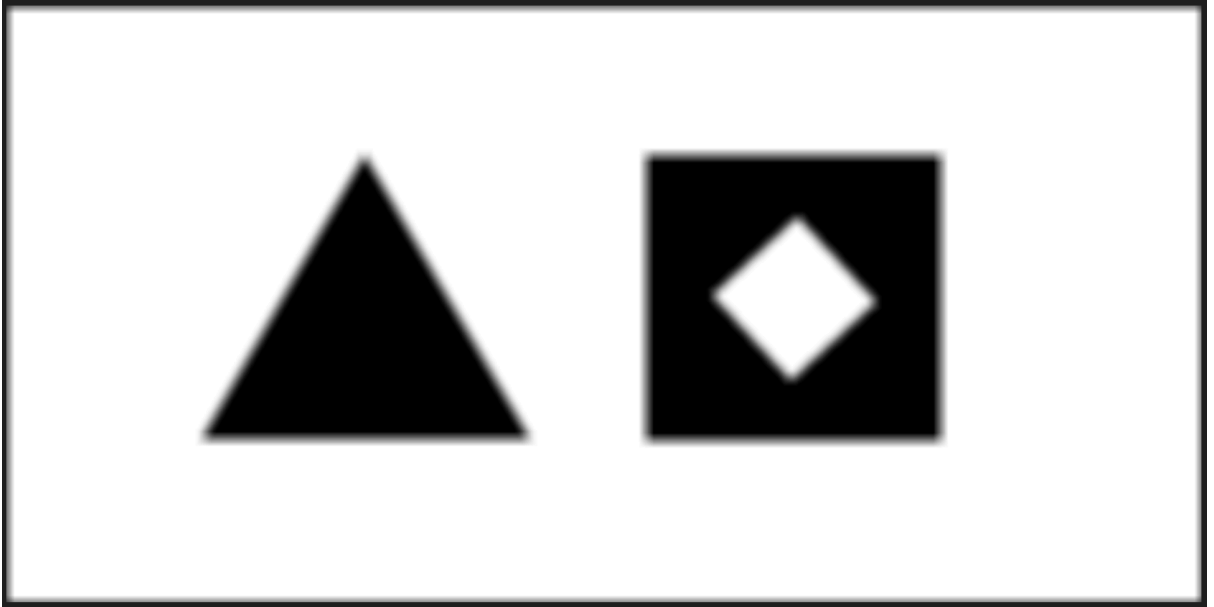
Logo de cara é possível perceber um embaçamento horizontal olhando para as bordas do quadrado, o que fica mais claro quando comparamos com o box 10x1 a seguir.

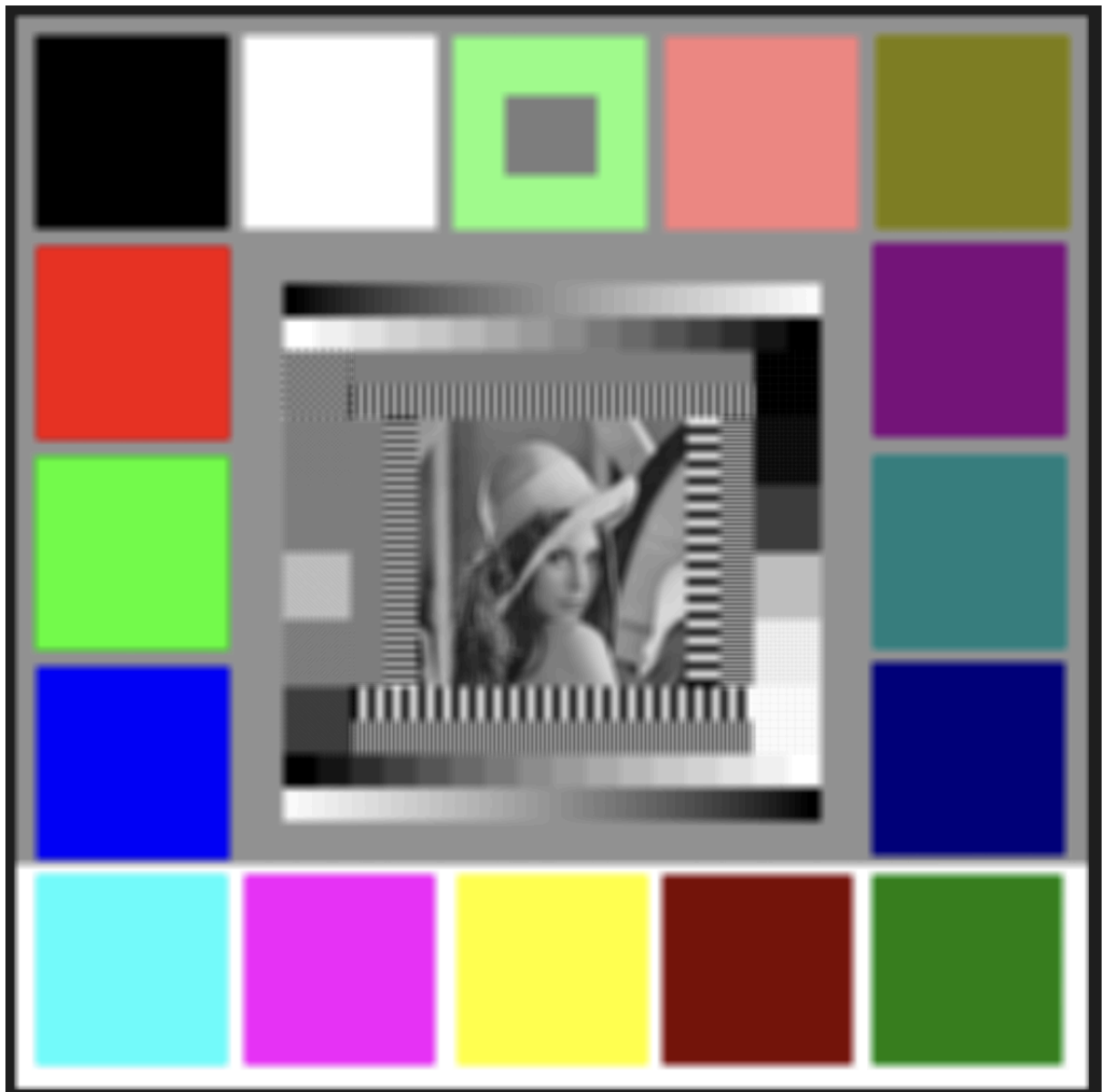
- Box10x1:



Aqui é possível perceber as laterais verticais do quadrado quase que perdendo suas bordas, o que testifica quando ao embaçamento vertical.

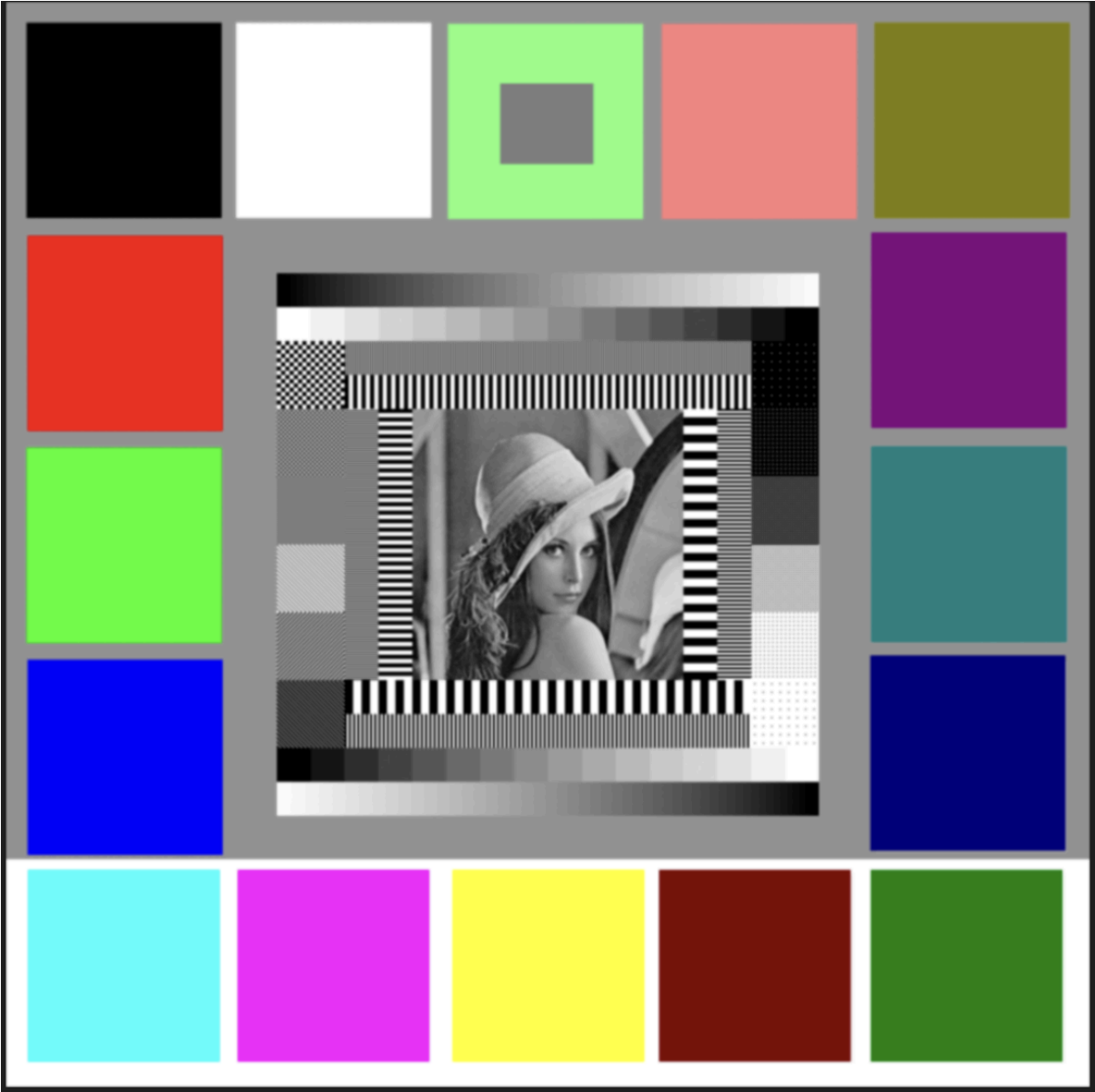
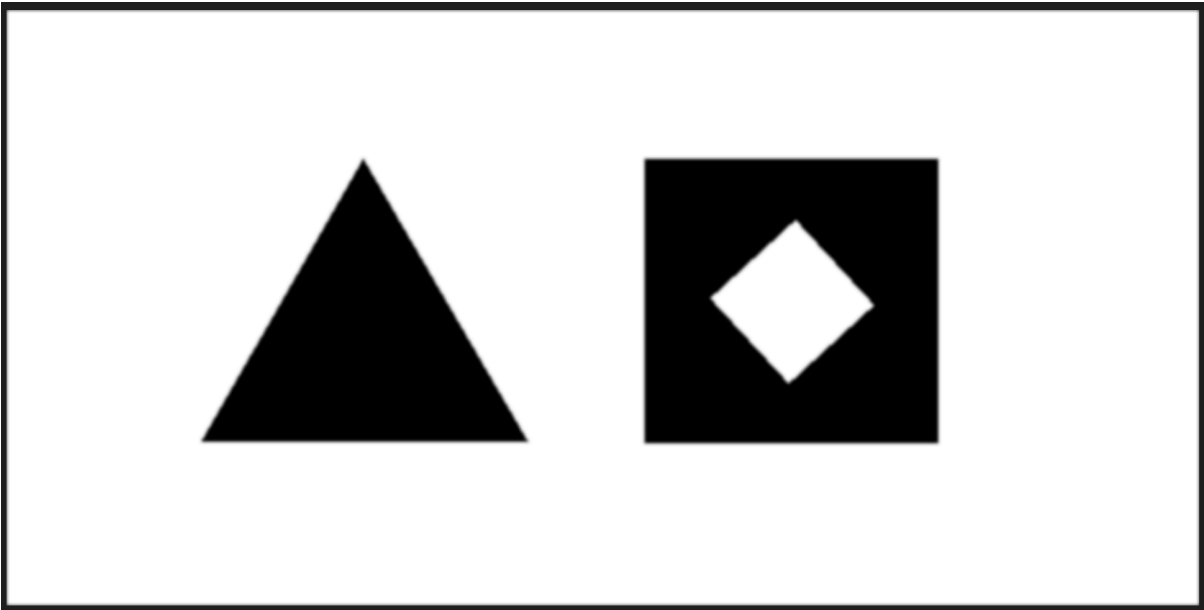
- box10x10:





Aqui é perceptível que não só as laterais do quadrado ficaram suavizadas, mas também a imagem como um todo ficou embaçada.

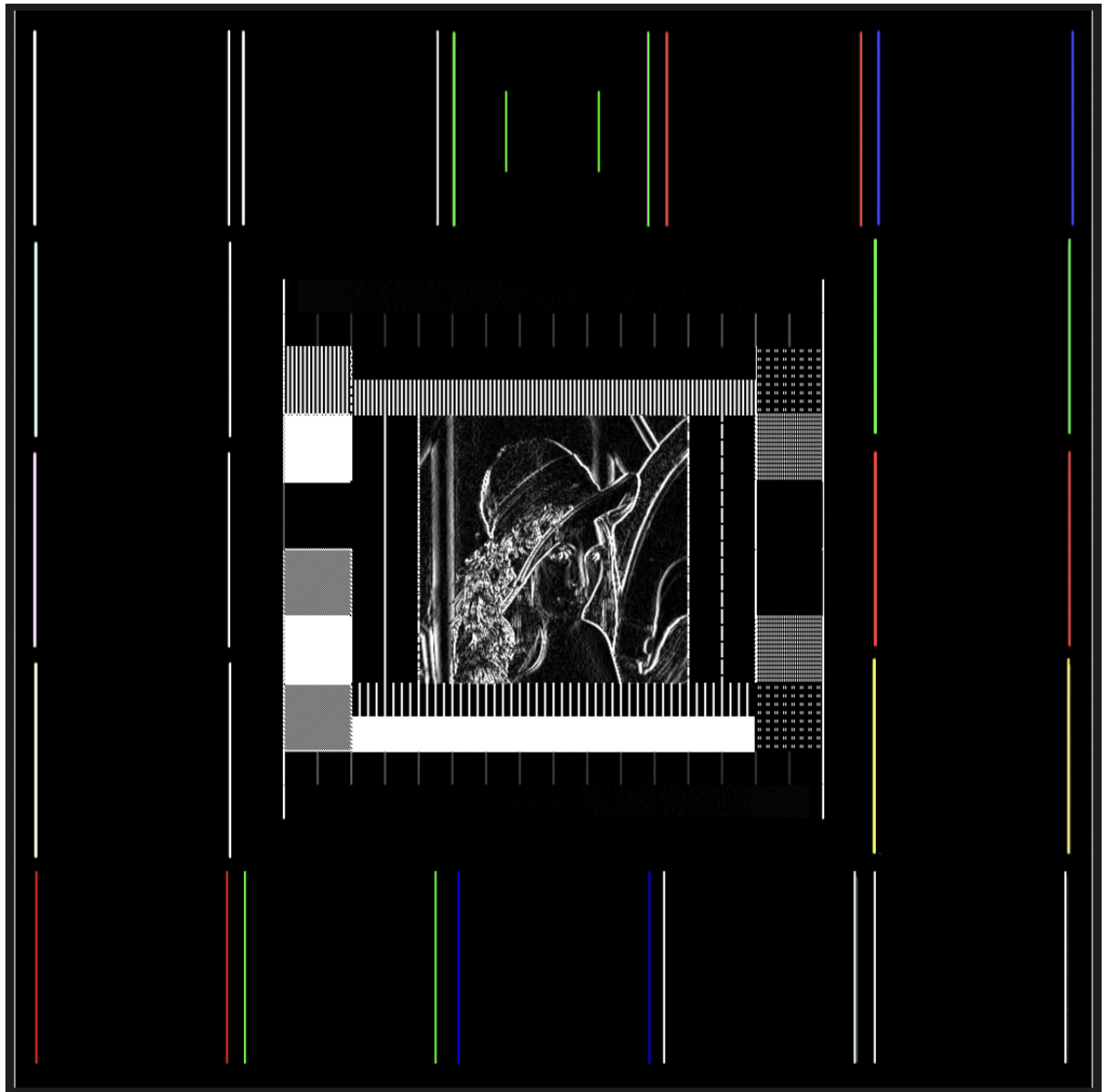
- gaussiano5x5:



Talvez aqui haja uma dificuldade maior de enxergar alguma diferença, mas tenho o sentimento que houve uma suavização sóbria na imagem, isto é: sem perda dos contornos e sem perda de contraste.

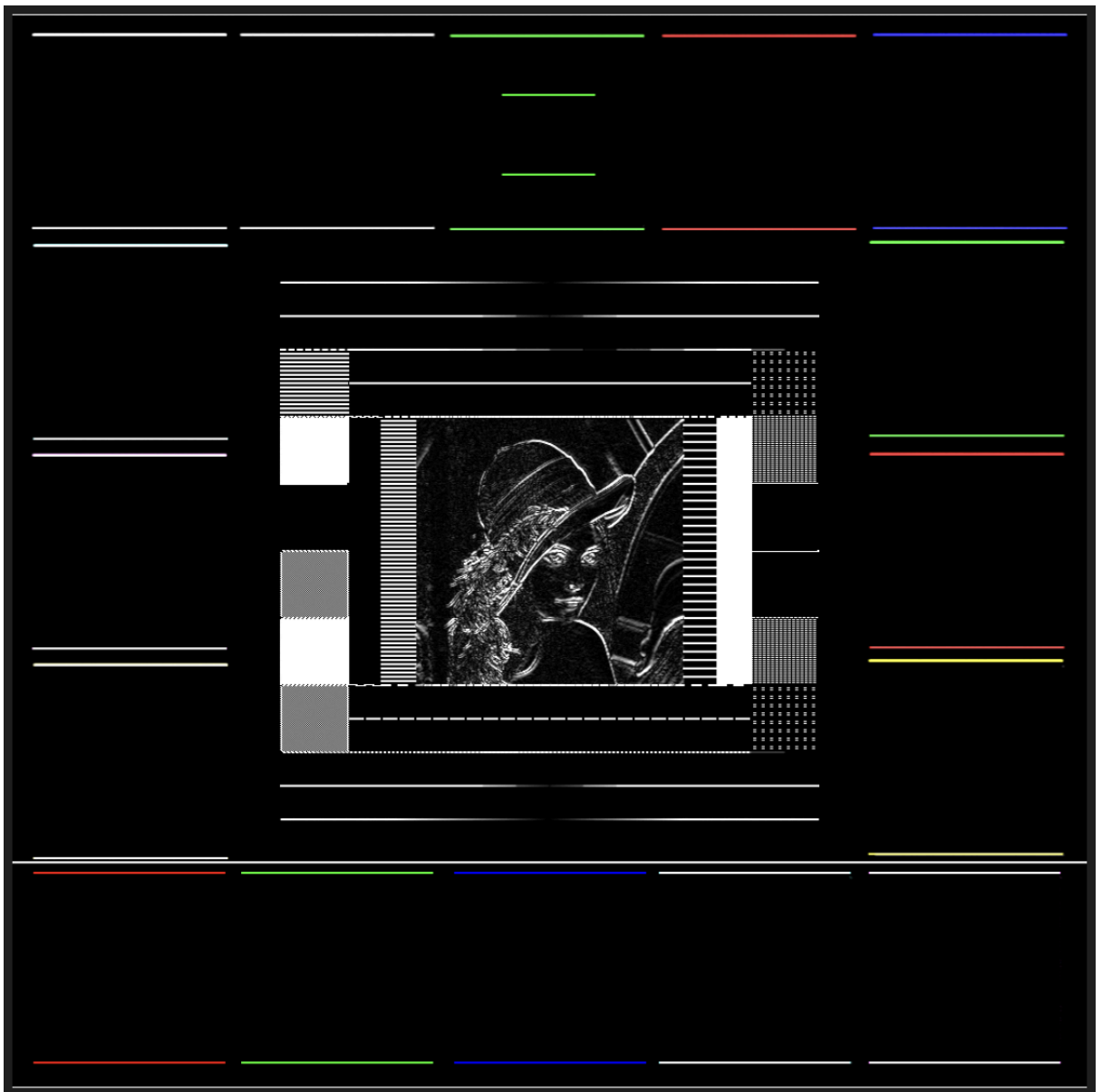
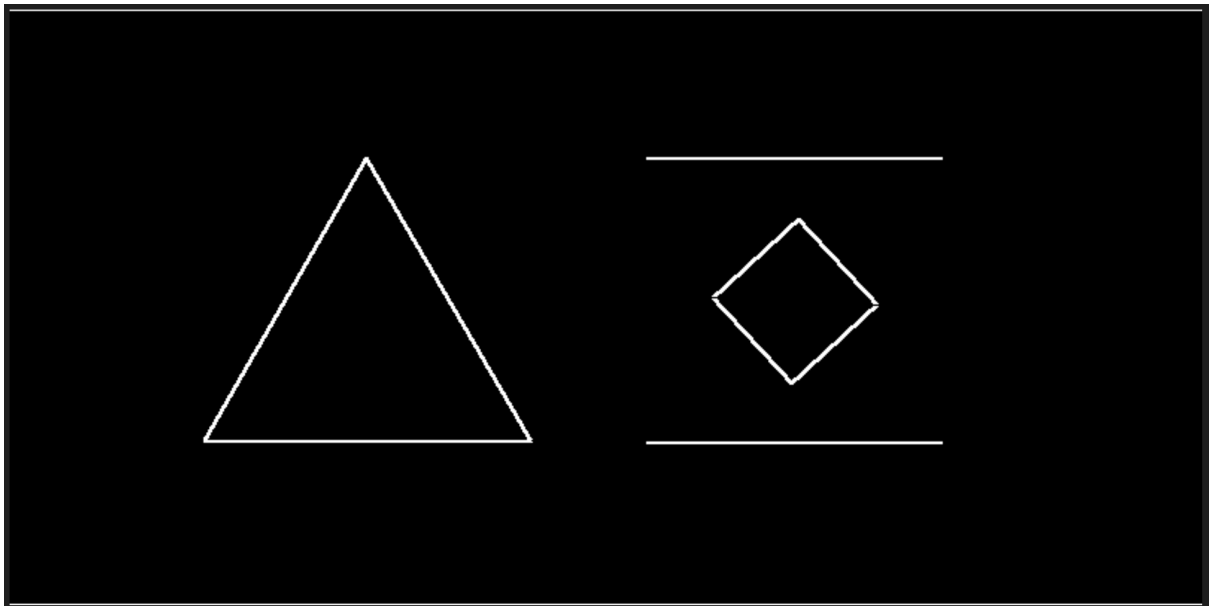
- sobel_v:





Acredito que o resultado é óbvio e esperado, os pixel que bordas foram mantidos ou intensificados e quanto os pixels uniformes aos seus vizinhos foram zerados, porém aqui isso é feito na vertical.

- sobel_h:



Agora temos o mesmo que o de cima porém as bordas horizontais foram zeradas.

Discussão

Durante o desenvolvimento surgiram dificuldades como erros de índices, geração de imagens totalmente pretas e a necessidade de normalização dos valores após a aplicação dos kernels. Esses problemas exigiram correções na lógica de acesso aos pixels e no tratamento dos limites da imagem.

Quanto ao desempenho, a abordagem em Go mostrou-se eficiente, mas o uso intensivo de goroutines(threads) em regiões pequenas da imagem trouxe sobrecarga desnecessária. Além disso, a técnica é limitada pela ausência de extensões nos contornos da imagem, o que pode afetar a qualidade do resultado em bordas.

Depois de muito tempo(horas) debugando o código e encontrei o erro:

```
if ix < 0 || ix >= dimensoes.Max.X || iy < 0 || iy >= dimensoes.Max.Y {  
    continue  
}
```

Antes, ao invés de tentar não extravasar a imagem(como está expresso acima na versão corrigida), eu estava tentando não extravasar o box da imagem sobre o qual estava o kernel naquela iteração.

Conclusão

Os objetivos do trabalho foram atingidos, permitindo a aplicação de diferentes kernels para processamento de imagens. A implementação em Go trouxe experiência prática com concorrência e manipulação de pixels, pois uso Go há algum tempo e goroutines, mutex e semáforos já estão se tornando naturais para mim, porém como a multiplicação das dimensões da imagem $M \times N$, essa abordagem se mostrou inadequada para a CPU, de fato agora pude entender realmente a importância das GPUs no processamento “concorrente” digital de imagens. Como contribuição, destacou-se o aprendizado sobre correlação, normalização e funcionamento de filtros no domínio espacial.