

1. O resultado em questão seria: [226, 166, 119, 55]. Dado que basta pegar o valor máximo da cor de um pixel, 255, e subtrair pelo seu próprio valor. Tendo assim o pixel invertido.
2. Imagem do peixe salva no repositório com o nome **inverted_fish.png**.
3. O cálculo foi feito originalmente através de um loop em python, obtendo o valor de: 32.760000000000005. Porém, o mesmo pode ser encontrado através deste calculo em qualquer calculadora: $53 \times -0,07 + 129 \times -0,45 + 127 \times 1,20 + 148 \times -0,25 + 174 \times -0,12$. Note que, os valores que resultam em 0, foram ignorados pois este foi feito manualmente. O processo de aplicação de kernel seria buscar o valor do pixel da imagem e o valor de pixel no kernel e multiplicá-los, após obter o resultado de cada um, basta somar todos.
4. Imagem do porco salva no repositório com o nome: **pig_result.png**.
5. Imagem do python salva no repositório com o nome: **sharpened_python.png**.
6. Imagem do construct salva no repositório com o nome: **edges_construct.png**.
A explicação do que cada kernel faz é relativamente simples. As duas servem para verificar uma mudança de intensidade entre os pixels da imagem, analisando os dois eixos, X e Y, por isso contém dois kernels. O resultado final da imagem é a combinação da aplicação dos dois kernels por toda imagem, dado que cada um por si só obtém resultados diferentes pois a orientação de aplicação é diferente.