Temat projektu

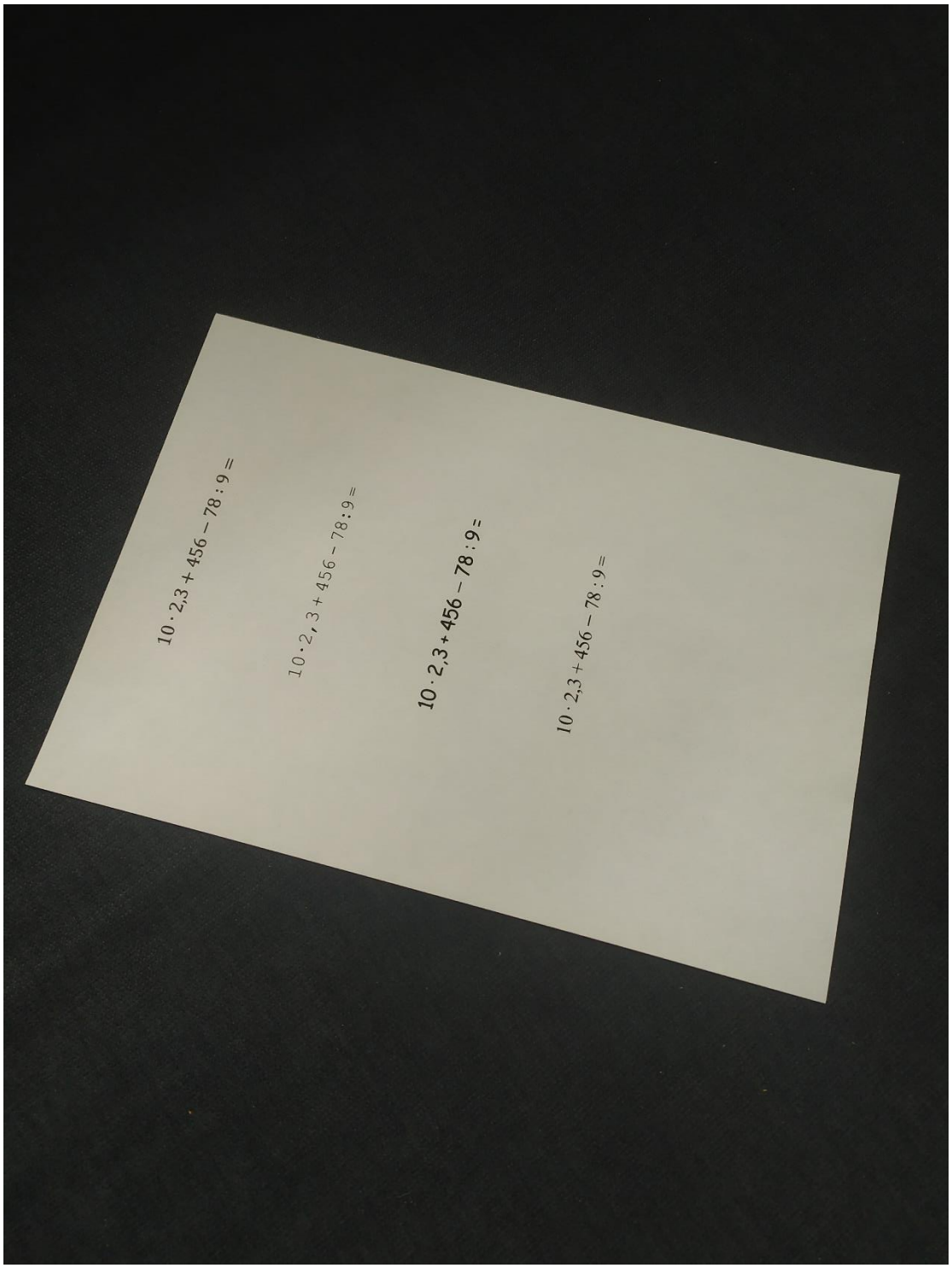
Projekt ma na celu wykrycie działań matematycznych na białej kartce. Znaki zawierają się w przedziale:

0 1 2 3 4 5 6 7 8 9 + − · : , =

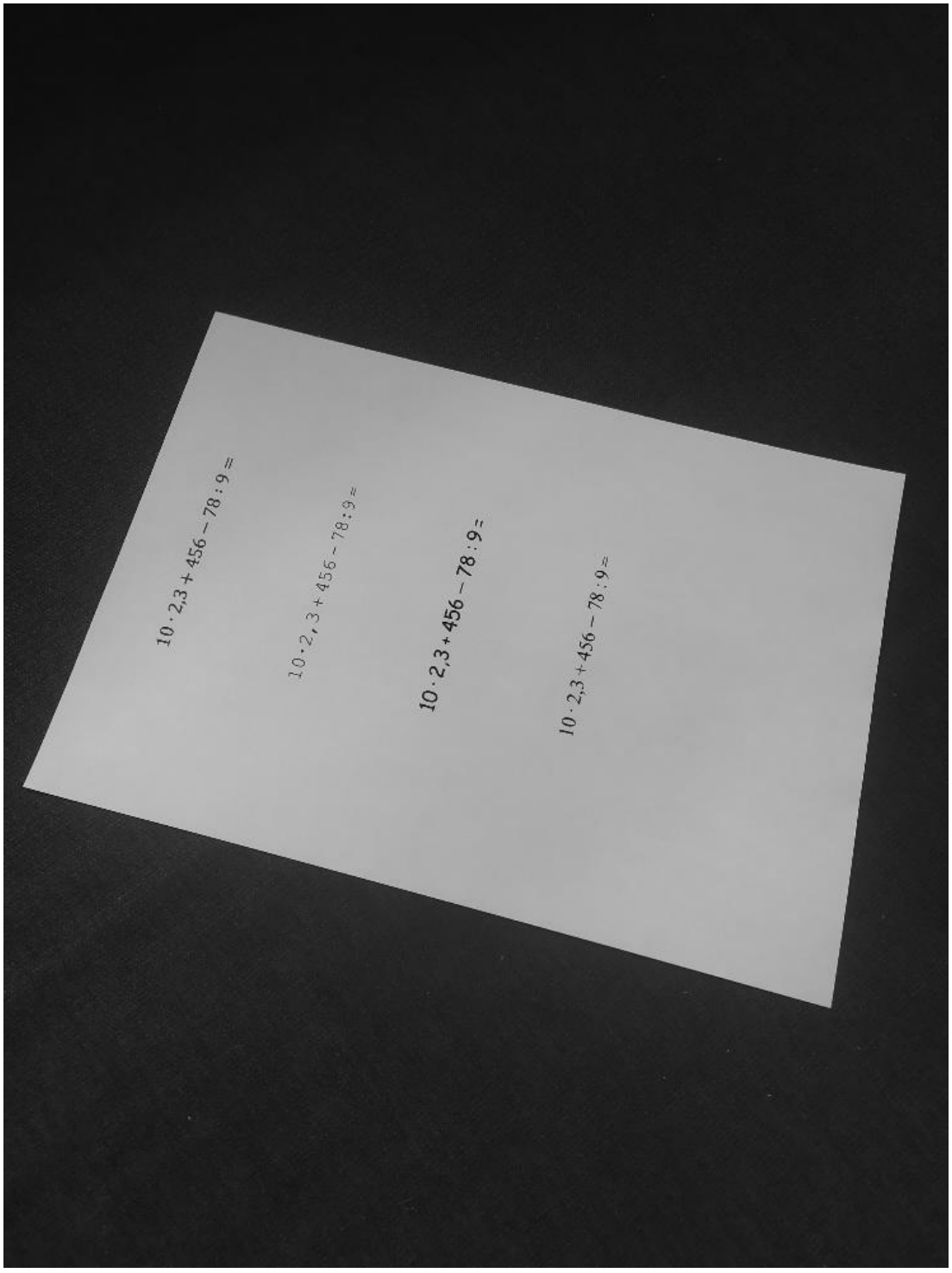
Kartka może być obrócona i pochylona – jedynym warunkiem jest, aby róg kartki będący najbliżej górnej krawędzi zdjęcia był dowolnym z górnych. Tło, na którym leży kartka powinno umiarkowanie być ciemne i jednolite.

Etapy realizacji problemu

- Oryginalne zdjęcie

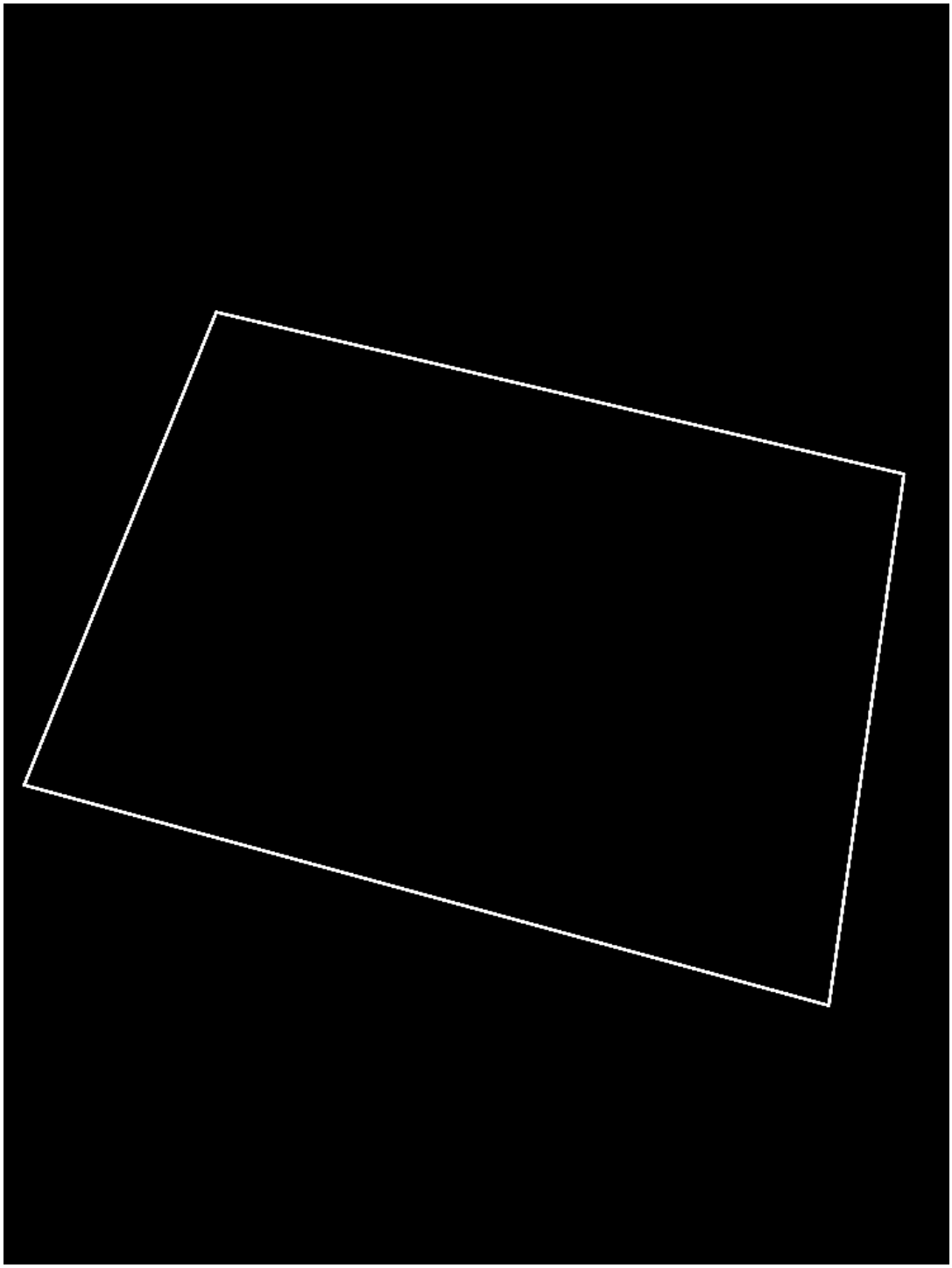


- Wczytanie zdjęcia w odcieniach szarości

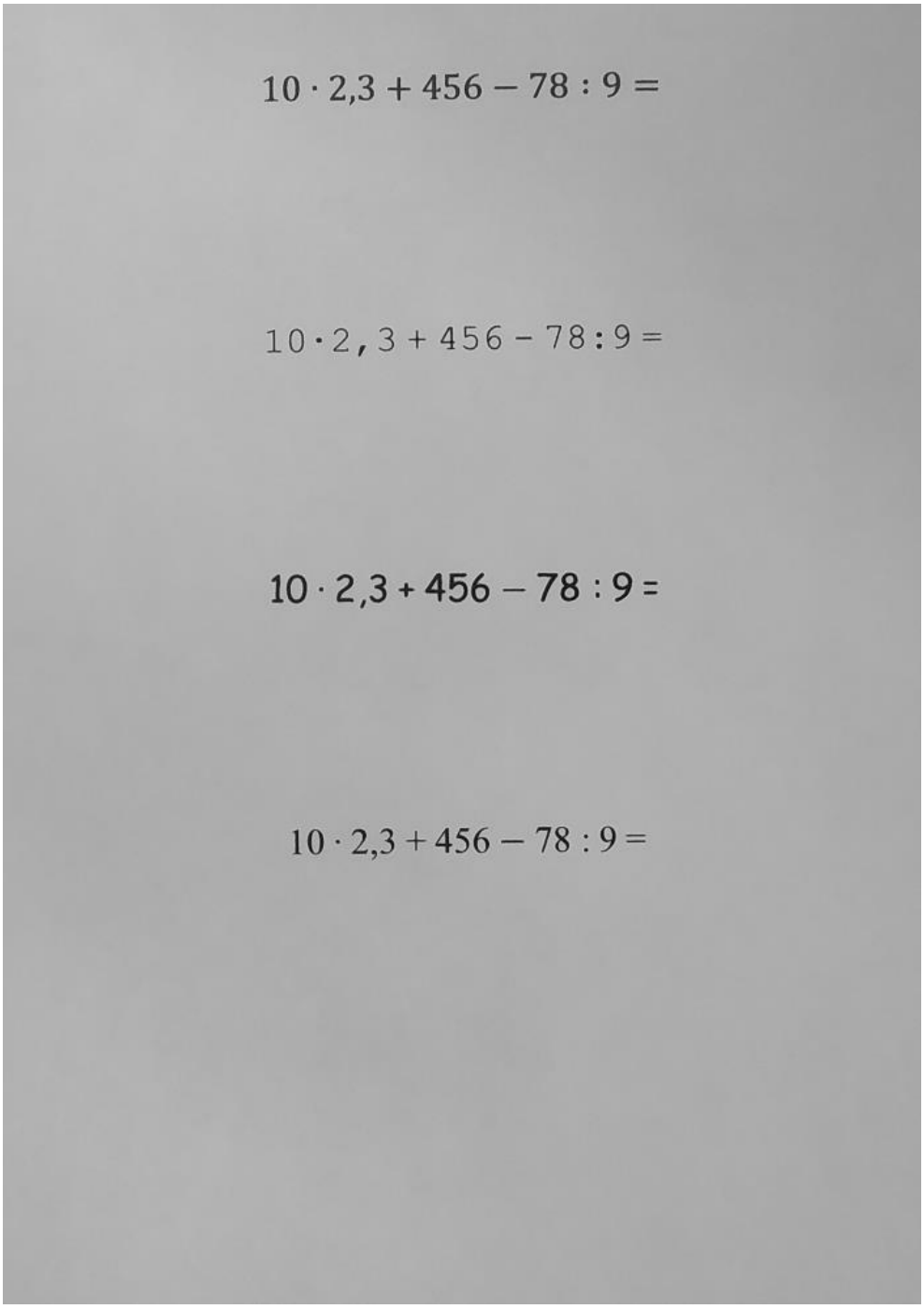


- Progowanie i filtracja szumu filtrem

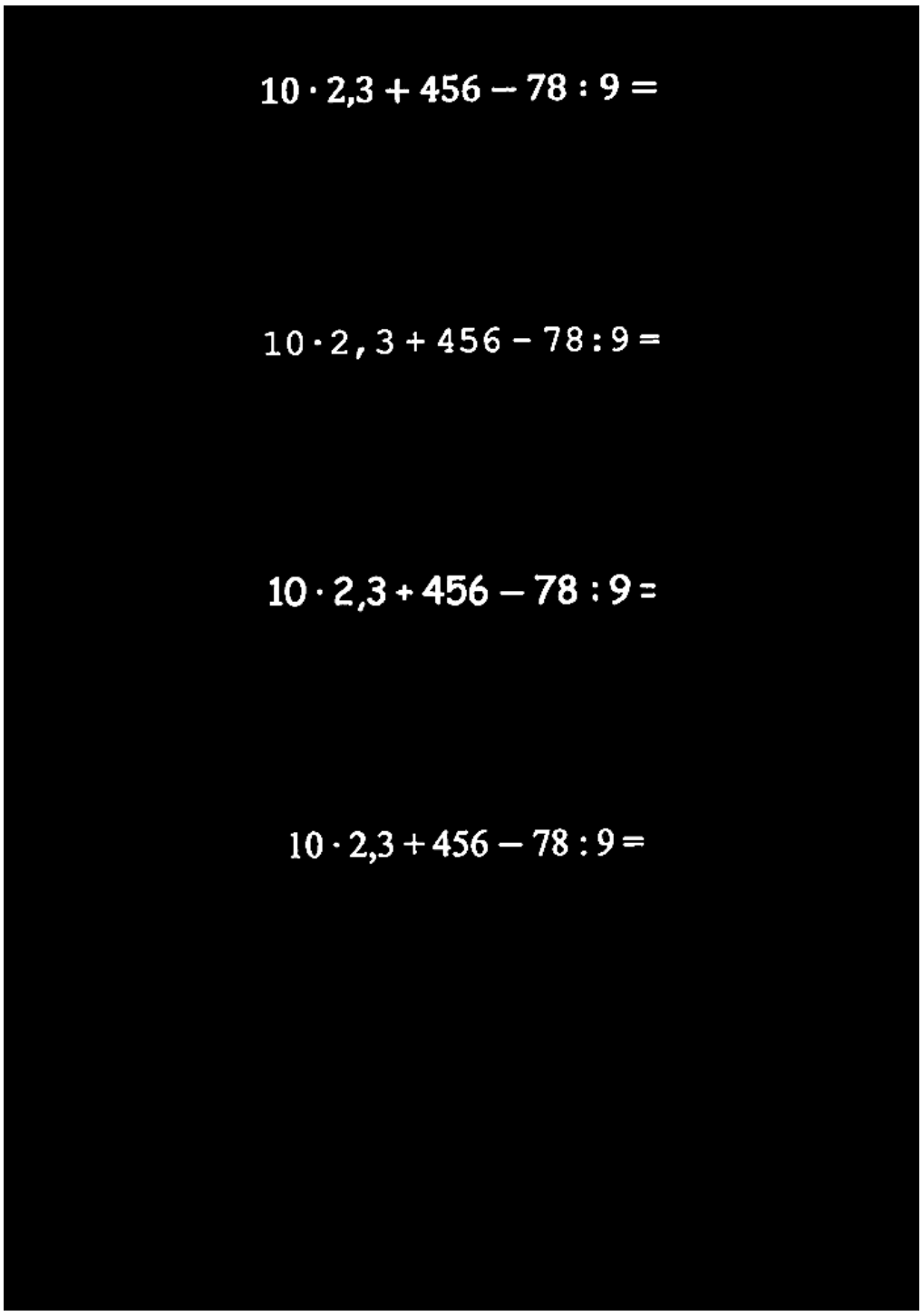
- Wykrycie konturów, wybranie największego powierzchnią oraz przybliżenie czworokątem – otrzymujemy współrzędne rogów



- Znalezione rogi są ułożone w kolejności przeciwnej do wskazówek zegara. Znaleziony zostaje najwyższy punkt, a następnie obliczane długości boków, do których należy ten punkt. Na tej podstawie możliwe jest zidentyfikowanie orientacji kartki i poprawa perspektywy.
-
- Zdjęcie w odcieniach szarości po poprawieniu perspektywy



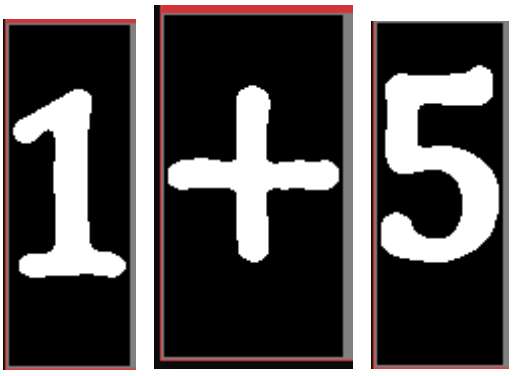
- Progowanie i filtracja jak poprzednio

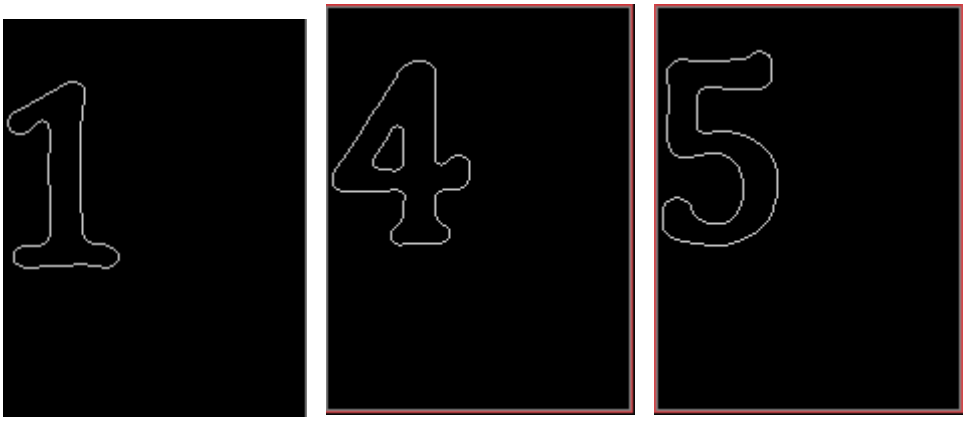


- Przeszukiwanie wierszami w poszukiwaniu białych pikseli. Na podstawie ich obecności wycinane są linijki z pojedynczymi równaniami.

- Następnie zostają wykryte kontury, posortowane od lewej do prawej oraz wycinane są pojedyncze znaki na podstawie skrajnych (lewo-prawo) punktów ich konturów. Wycięty znak jest zamalowany, aby uniknąć podwajania się znaku : lub =. Pionowe linie służą jedynie poglądowi

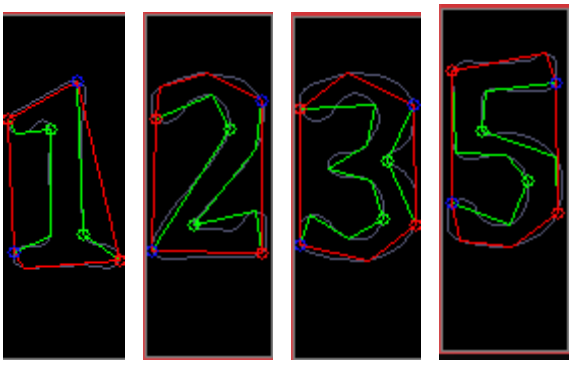
- Przykładowe wycięte znaki i ich kontury





Rozpoznawanie znaków wygląda następująco:

- Wykrycie konturów znaku
- Usunięcie małych konturów będących szumami
- Policzenie konturów tworzących znak
  - Jeśli są trzy to jest to liczba „8”
  - Jeśli są dwa
    - Mają małą powierzchnię (: =):
      porównywanie ze wzorcami za pomocą cv2.mathShapes()
    - Mają dużą powierzchnię (0,4,6,9):
      Znalezienie środka wewnętrznego konturu i na tej podstawie sklasyfikowanie znaku
  - Jeśli jest jeden
    - Wysokość konturu jest mała (· - ,):
      porównywanie ze wzorcami za pomocą cv2.mathShapes()
    - Wysokość jest duża (1,2,3,5,7, +):
      Przybliżenie wielokątem, a następnie stworzenie otoczki wypukłej. Obliczenie ilości „defektów” wielokąta w stosunku do otoczki.
      - 4 defekty – „+”
      - 1 defekt – „7”
      - 2 defekty:
        Porównanie położenia defektów względem wymiarów wielokąta – rozpoznanie „2” oraz „5”. Pozostałe znaki (1, 3) rozpoznaje się wykrywając linie Hougha. Jeśli istnieje to „1”. Jeśli nie to „3”
      -
- Przykładowe otoczki i wielokąty



Wynik rozpoznania znaków: