МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**ЗВІТ**

**з лабораторної роботи № 5**

**«Обробка даних у застосунках Android»**

**Виконав:**

студент 3-го курсу, групи КП-83, спеціальності 121 – Інженерія програмного забезпечення

*Мар’яненко Роман Геннадійович*

**Перевірив:**

к. т. н, старший викладач

*Хайдуров Владислав Володимирович*

Київ – 2020

**ЗМІСТ**

**ВСТУП**……………………………………………………………………………….**3**

1. Завдання до лабораторної роботи..………………………………………………4

2. Короткі теоретичні відомості…………………………………………………….5

3. Програмна реалізація задачі……………………………………………………...6

4. Результати роботи програми…………………………………………………....17

**ВИСНОВКИ**………………………………………………………………………..19

**ВСТУП**

Під час виконання лабораторної роботи ми ознайомитись із основними принципами обробки графічних даних з використанням основних методів і алгоритмів цифрової обробки зображень.

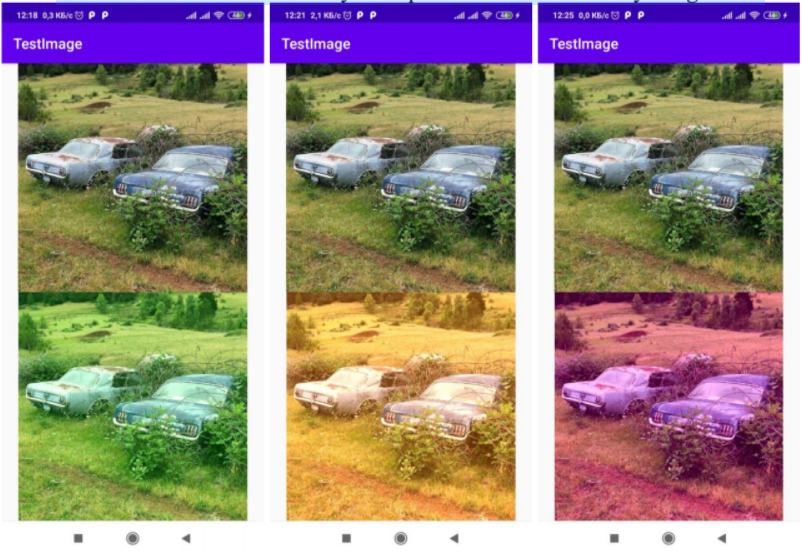
**Завдання до лабораторної роботи**

1. Ознайомитись із усіма теоретичними відомостями до лабораторної роботи.

2. Створити мобільний застосунок, який виконує інвертування кольорової палітри попередньо підготовленого зображення, що міститься у папці drawable проекту програмного застосунку. Вивести вхідне та отримане зображення в об’єкти типу ImageView.

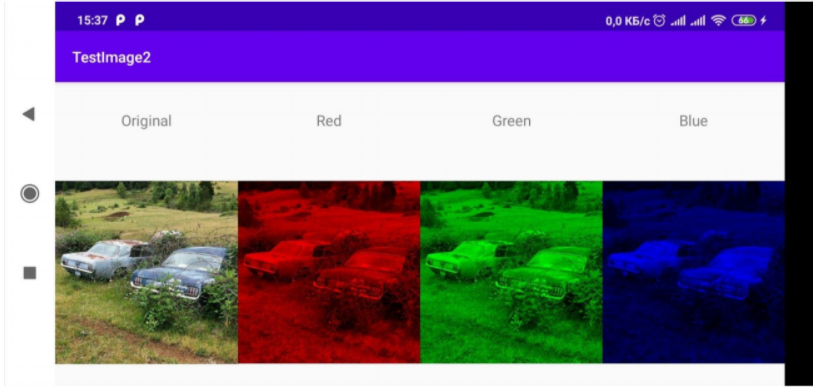


3. Створити мобільний застосунок, який для деякої (однієї з трьох) компонент виконує зміну, наприклад, до кожного пікселя додає якесь постійне значення. Вивести усі зображення в об’єкти типу ImageView.



4. Створити мобільний застосунок, за допомогою якого виконується

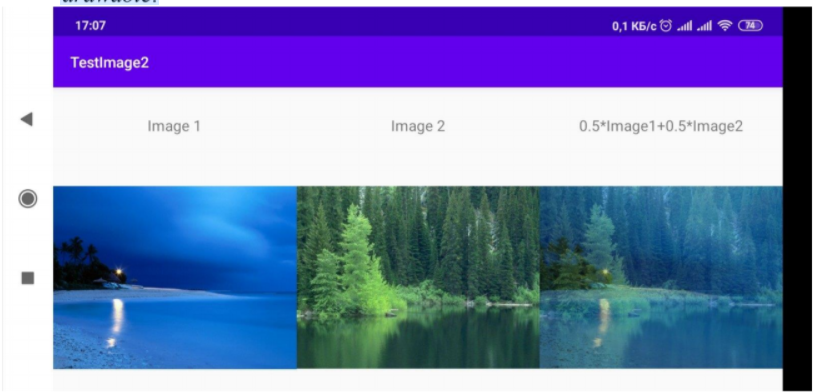
розбивка зображення на компоненти Red, Green та Blue.



5. Створити мобільний застосунок, за допомогою якого виконується «злиття» двох зображень у певних пропорціях (долях від одиниці, яка береться за 100%). Наприклад, якщо «вклад» першого зображення у нове складає 0.4, то «вклад» другого – 1 - 0.4 = 0.6. У загальному «вклад» першого зображення позначимо через α, а «вклад» другого – (1 − α). Тоді шукане зображення можна записати у наступному вигляді:

***Imagenew = α ∙ Image1 + (1 − α)Image2.***

Вхідні файли також попередньо підготувати та додати їх до папки drawable.



6. Створти мобільний застосунок, що виконує фільтрацію вхідного зображення з використанням матричних фільтрів, що описані у лабораторній роботі (фільтр розмиття, фільтр поліпшення чіткості, медіанний фільтр, фільтр ерозії і нарощування та фільтр Собеля).

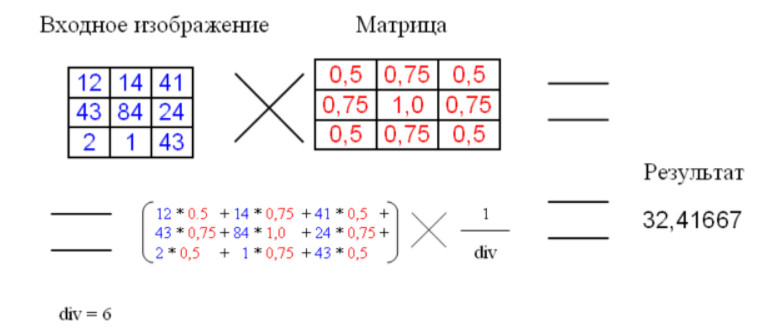
7. Створти мобільний застосунок, що виконує вбудовування водяного знаку зображення, що попередньо підготовлені та містяться в папці drawable мобільного застосунку. Для процесу вбудовування водяний знак перетворити до бінарного (чорно-білого, не сірого!). Використати для вбудовування метод найменшого значущого біта. Параметр «номер бітової площини» (натуральне число, менше за 9) вивести на форму (Activity) для відображення різниці в результатах вбудовування. Водяний знак вбудовувати у канал Blue. Передбачити випадок різних розмірів вхідного зображення (контейнера) та водяного знаку. Якщо вхідне зображення більше за водяний знак, виконати вбудовування циклічно та періодично для всього вхідного зображення (замостити вхідне зображення водяним знаком). Якщо вхідне зображення менше за водяний знак, вбудувати частину водяного знаку для всього вхідного зображення.

8. Додатковий бал передбачається з встворення методу, який виконує вилучення вонядного знаку із заповненого конейнега (зображення з водяним знаком) для завдання 7 даної лабораторної роботи.

9. Усі завдання лабораторної роботи за бажанням можна виконати в одному багатовіконному мобільному додатку.

**Короткі теоретичні відомості**

Фільтрів, що використовують матрицю згортки багато. Матриця згортки – це матриця коефіцієнтів, яка «множиться» на значення пікселів зображення для отримання необхідного результату. Нижче представлено застосування матриці згортки:



div – це коефіцієнт нормування, для того щоб середня інтенсивність залишалася зрадою. У прикладі матриця має розмір 3x3, хоча розмір може бути і більше.

Фільтри нарощування і ерозія служать для отримання морфологічного розширення або звуження відповідно. Простіше кажучи, для зображень це означає вибір пікселя з максимальною або мінімальною інтенсивністю з околиці.

Оператор Собеля використовується в обробці зображень для виділення границь. Це дискретний диференціальний оператор, що обчислює наближене значення градієнта чи норми градієнта для яскравості зображення. Оператор Собеля базується на згортці зображення невеликими сепарабельними цілочисельними фільтрами в вертикальному та горизонтальному напрямках. Хоча, апроксимація градієнта досить груба, особливо на високочастотних ділянках зображення.

**Програмна реалізація задачі**

|  |
| --- |
| layout/activity\_main.xml |
| <?xml version="1.0" encoding="utf-8"?>  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="@color/colorBackground">  <LinearLayout  android:layout\_width="0dp"  android:layout\_height="0dp"  android:orientation="vertical"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent">  <Space  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="3" />  <Button  android:id="@+id/inverse\_image"  style="@style/main\_activity\_button"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="@string/task1\_button" />  <Space  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1" />  <Button  android:id="@+id/apply\_a\_color\_filter"  style="@style/main\_activity\_button"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="@string/task2\_button" />  <Space  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1" />  <Button  android:id="@+id/split\_by\_colors"  style="@style/main\_activity\_button"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="@string/task3\_button" />  <Space  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1" />  <Button  android:id="@+id/image\_overlay"  style="@style/main\_activity\_button"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="@string/task4\_button" />  <Space  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1" />  <Button  android:id="@+id/filters"  style="@style/main\_activity\_button"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="@string/task5\_button" />  <Space  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="3" />  </LinearLayout>  </androidx.constraintlayout.widget.ConstraintLayout> |

|  |
| --- |
| layout/two\_images\_layout.xml |
| <?xml version="1.0" encoding="utf-8"?>  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".MainActivity">  <LinearLayout  android:layout\_width="0dp"  android:layout\_height="0dp"  android:orientation="vertical"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent">  <ImageView  android:id="@+id/original"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:contentDescription="@string/original\_photo\_placeholder"  tools:srcCompat="@tools:sample/avatars" />  <ImageView  android:id="@+id/modified"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:contentDescription="@string/modefied\_photo\_placeholder"  tools:srcCompat="@tools:sample/avatars" />  </LinearLayout>  </androidx.constraintlayout.widget.ConstraintLayout> |

|  |
| --- |
| layout/activity\_split\_by\_colors.java |
| <?xml version="1.0" encoding="utf-8"?>  <androidx.constraintlayout.widget.ConstraintLayout  xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent">  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical">  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:baselineAligned="true"  android:orientation="horizontal">  <androidx.constraintlayout.widget.ConstraintLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:baselineAligned="false"  android:orientation="vertical">  <ImageView  android:id="@+id/original"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:contentDescription="@string/original\_photo\_placeholder"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent"  tools:srcCompat="@tools:sample/avatars" />  <TextView  android:id="@+id/textView"  style="@style/image\_caption"  android:text="@string/original\_photo\_caption"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/original" />  </androidx.constraintlayout.widget.ConstraintLayout>  <androidx.constraintlayout.widget.ConstraintLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:orientation="vertical">  <ImageView  android:id="@+id/red"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:contentDescription="@string/original\_photo\_placeholder"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent"  tools:srcCompat="@tools:sample/avatars" />  <TextView  android:id="@+id/textView1"  style="@style/image\_caption"  android:text="@string/red\_photo\_caption"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/red" />  </androidx.constraintlayout.widget.ConstraintLayout>  </LinearLayout>  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:baselineAligned="true"  android:orientation="horizontal">  <androidx.constraintlayout.widget.ConstraintLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:orientation="vertical">  <ImageView  android:id="@+id/green"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:contentDescription="@string/original\_photo\_placeholder"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent"  tools:srcCompat="@tools:sample/avatars" />  <TextView  android:id="@+id/textView2"  style="@style/image\_caption"  android:text="@string/green\_photo\_caption"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/green" />  </androidx.constraintlayout.widget.ConstraintLayout>  <androidx.constraintlayout.widget.ConstraintLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:orientation="vertical"  tools:layout\_editor\_absoluteX="206dp">  <ImageView  android:id="@+id/blue"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:contentDescription="@string/original\_photo\_placeholder"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent"  tools:srcCompat="@tools:sample/avatars" />  <TextView  android:id="@+id/textView3"  style="@style/image\_caption"  android:text="@string/blue\_photo\_caption"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/blue" />  </androidx.constraintlayout.widget.ConstraintLayout>  </LinearLayout>  </LinearLayout>  </androidx.constraintlayout.widget.ConstraintLayout> |

|  |
| --- |
| layout/activity\_image\_overlay.xml |
| <?xml version="1.0" encoding="utf-8"?>  <androidx.constraintlayout.widget.ConstraintLayout  xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent">  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:orientation="vertical"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent">  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:gravity="center\_vertical"  android:orientation="horizontal">  <Button  android:id="@+id/decrease\_alpha"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="@string/decrease\_alpha" />  <TextView  android:id="@+id/alpha"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:gravity="center\_horizontal|center\_vertical"  android:text="@string/alpha\_placeholder"  android:textColor="#000000"  android:textSize="24sp"  android:textStyle="bold" />  <Button  android:id="@+id/increase\_alpha"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="@string/increase\_alpha" />  </LinearLayout>  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="1"  android:gravity="center\_vertical"  android:orientation="vertical">  <ImageView  android:id="@+id/original1"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  tools:srcCompat="@tools:sample/avatars" />  <ImageView  android:id="@+id/modified"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  tools:srcCompat="@tools:sample/avatars" />  <ImageView  android:id="@+id/original2"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  tools:srcCompat="@tools:sample/avatars" />  </LinearLayout>  </LinearLayout>  </androidx.constraintlayout.widget.ConstraintLayout> |

|  |
| --- |
| layout/activity\_filters.xml |
| <?xml version="1.0" encoding="utf-8"?>  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".FiltersActivity">  <LinearLayout  android:layout\_width="409dp"  android:layout\_height="729dp"  android:orientation="vertical"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent">  <ImageView  android:id="@+id/view"  android:layout\_width="match\_parent"  android:layout\_height="446dp"  android:scaleType="centerInside"  tools:srcCompat="@tools:sample/avatars" />  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="vertical">  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="horizontal">  <Button  android:id="@+id/original"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="@string/task5\_original\_button" />  <Button  android:id="@+id/blur"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="@string/task5\_blur\_button" />  <Button  android:id="@+id/sharpness"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="@string/task5\_sharpness\_button" />  </LinearLayout>  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="horizontal">  <Button  android:id="@+id/median"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="@string/task5\_median\_button" />  <Button  android:id="@+id/growingUp"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="@string/task5\_growingUp\_button" />  <Button  android:id="@+id/sobel"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="@string/task5\_sobel\_button" />  </LinearLayout>  </LinearLayout>  </LinearLayout>  </androidx.constraintlayout.widget.ConstraintLayout> |

|  |
| --- |
| MainActivity.java |
| package com.example.lab5;  import androidx.appcompat.app.AppCompatActivity;  import android.content.Intent;  import android.graphics.Bitmap;  import android.graphics.BitmapFactory;  import android.os.Bundle;  import android.view.View;  import android.widget.Button;  public class MainActivity extends AppCompatActivity {  Button inverseImage, applyAColorFilter, splitByColors, imageOverlay, filters, task5, task6;  public static final int maxColorIntensity = 255;  static Bitmap originalMap;  public static int mapWidth, mapHeight;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  initButtons();  initBitmap();  inverseImage.setOnClickListener(getListener(InverseImageActivity.class));  applyAColorFilter.setOnClickListener(getListener(ApplyAColorFilterActivity.class));  splitByColors.setOnClickListener(getListener(SplitByColorsActivity.class));  imageOverlay.setOnClickListener(getListener(ImageOverlayActivity.class));  filters.setOnClickListener(getListener(FiltersActivity.class));  }  private void initBitmap() {  originalMap = BitmapFactory.decodeResource(getResources(), R.drawable.garold);  mapWidth = originalMap.getWidth();  mapHeight = originalMap.getHeight();  }  private void initButtons() {  inverseImage = findViewById(R.id.inverse\_image);  applyAColorFilter = findViewById(R.id.apply\_a\_color\_filter);  splitByColors = findViewById(R.id.split\_by\_colors);  imageOverlay = findViewById(R.id.image\_overlay);  filters = findViewById(R.id.filters);  }  private View.OnClickListener getListener(final Class<?> cl) {  return new View.OnClickListener() {  @Override  public void onClick(View v) {  Intent intent = new Intent();  intent.setClass(MainActivity.this, cl);  startActivity(intent);  }  };  }  } |

|  |
| --- |
| InverseImageActivity.java |
| package com.example.lab5;  import android.graphics.Bitmap;  import android.graphics.BitmapFactory;  import android.graphics.Color;  import android.os.Bundle;  import android.widget.ImageView;  import androidx.appcompat.app.AppCompatActivity;  public class InverseImageActivity extends AppCompatActivity {  ImageView originalView, invertedView;  Bitmap invertedMap;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.two\_images\_layout);  initViews();  int[] pixels = new int[MainActivity.mapWidth \* MainActivity.mapHeight];  MainActivity.originalMap.getPixels(pixels, 0, MainActivity.mapWidth, 0, 0, MainActivity.mapWidth, MainActivity.mapHeight);  int[] invertedPixels = invertPixels(pixels);  invertedMap = Bitmap.createBitmap(MainActivity.mapWidth, MainActivity.mapHeight, Bitmap.Config.ARGB\_8888);  invertedMap.setPixels(invertedPixels, 0, MainActivity.mapWidth, 0, 0, MainActivity.mapWidth, MainActivity.mapHeight);  originalView.setImageBitmap(MainActivity.originalMap);  invertedView.setImageBitmap(invertedMap);  }  private void initViews() {  originalView = findViewById(R.id.original);  invertedView = findViewById(R.id.modified);  }  private int[] invertPixels(int[] pixels) {  for (int i = 0; i < pixels.length; i++) {  int alpha = Color.alpha(pixels[i]);  int red = MainActivity.maxColorIntensity - Color.red(pixels[i]);  int green = MainActivity.maxColorIntensity - Color.green(pixels[i]);  int blue = MainActivity.maxColorIntensity - Color.blue(pixels[i]);  pixels[i] = Color.argb(alpha, red, green, blue);  }  return pixels;  }  } |

|  |
| --- |
| ApplyAColorFilterActivity.java |
| package com.example.lab5;  import android.graphics.Bitmap;  import android.graphics.BitmapFactory;  import android.graphics.Color;  import android.os.Bundle;  import android.widget.ImageView;  import androidx.appcompat.app.AppCompatActivity;  public class ApplyAColorFilterActivity extends AppCompatActivity {  public static final int additionalIntensity = 80;  ImageView originalView, filteredView;  Bitmap filteredMap;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.two\_images\_layout);  initViews();  int[] pixels = new int[MainActivity.mapWidth \* MainActivity.mapHeight];  MainActivity.originalMap.getPixels(pixels, 0, MainActivity.mapWidth, 0, 0, MainActivity.mapWidth, MainActivity.mapHeight);  int[] filteredPixels = filterPixels(pixels);  filteredMap = Bitmap.createBitmap(MainActivity.mapWidth, MainActivity.mapHeight, Bitmap.Config.ARGB\_8888);  filteredMap.setPixels(filteredPixels, 0, MainActivity.mapWidth, 0, 0, MainActivity.mapWidth, MainActivity.mapHeight);  originalView.setImageBitmap(MainActivity.originalMap);  filteredView.setImageBitmap(filteredMap);  }  private void initViews() {  originalView = findViewById(R.id.original);  filteredView = findViewById(R.id.modified);  }  private int[] filterPixels(int[] pixels) {  for (int i = 0; i < pixels.length; i++) {  int alpha = Color.alpha(pixels[i]);  int red = Color.red(pixels[i]) + additionalIntensity;  int green = Color.green(pixels[i]);  int blue = Color.blue(pixels[i]) + additionalIntensity;  red = Math.min(red, MainActivity.maxColorIntensity);  blue = Math.min(blue, MainActivity.maxColorIntensity);  pixels[i] = Color.argb(alpha, red, green, blue);  }  return pixels;  }  } |

|  |
| --- |
| SplitByColorsActivity.java |
| package com.example.lab5;  import android.graphics.Bitmap;  import android.graphics.Color;  import android.os.Bundle;  import android.widget.ImageView;  import androidx.appcompat.app.AppCompatActivity;  public class SplitByColorsActivity extends AppCompatActivity {  ImageView originalView, redView, greenView, blueView;  Bitmap redMap, greenMap, blueMap;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_split\_by\_colors);  initViews();  redMap = Bitmap.createBitmap(MainActivity.mapWidth, MainActivity.mapHeight, Bitmap.Config.ARGB\_8888);  greenMap = Bitmap.createBitmap(MainActivity.mapWidth, MainActivity.mapHeight, Bitmap.Config.ARGB\_8888);  blueMap = Bitmap.createBitmap(MainActivity.mapWidth, MainActivity.mapHeight, Bitmap.Config.ARGB\_8888);  int[] pixels = new int[MainActivity.mapWidth \* MainActivity.mapHeight];  MainActivity.originalMap.getPixels(pixels, 0, MainActivity.mapWidth, 0, 0, MainActivity.mapWidth, MainActivity.mapHeight);  int[] redPixels = getRedPixels(pixels);  int[] greenPixels = getGreenPixels(pixels);  int[] bluePixels = getBluePixels(pixels);  redMap.setPixels(redPixels, 0, MainActivity.mapWidth, 0, 0, MainActivity.mapWidth, MainActivity.mapHeight);  greenMap.setPixels(greenPixels, 0, MainActivity.mapWidth, 0, 0, MainActivity.mapWidth, MainActivity.mapHeight);  blueMap.setPixels(bluePixels, 0, MainActivity.mapWidth, 0, 0, MainActivity.mapWidth, MainActivity.mapHeight);  originalView.setImageBitmap(MainActivity.originalMap);  redView.setImageBitmap(redMap);  greenView.setImageBitmap(greenMap);  blueView.setImageBitmap(blueMap);  }  private void initViews() {  originalView = findViewById(R.id.original);  redView = findViewById(R.id.red);  greenView = findViewById(R.id.green);  blueView = findViewById(R.id.blue);  }  private int[] getRedPixels(int[] pixels) {  int[] newPixels = new int[pixels.length];  for (int i = 0; i < pixels.length; i++) {  int alpha = Color.alpha(pixels[i]);  int red = Color.red(pixels[i]);  newPixels[i] = Color.argb(alpha, red, 0, 0);  }  return newPixels;  }  private int[] getGreenPixels(int[] pixels) {  int[] newPixels = new int[pixels.length];  for (int i = 0; i < pixels.length; i++) {  int alpha = Color.alpha(pixels[i]);  int green = Color.green(pixels[i]);  newPixels[i] = Color.argb(alpha, 0, green, 0);  }  return newPixels;  }  private int[] getBluePixels(int[] pixels) {  int[] newPixels = new int[pixels.length];  for (int i = 0; i < pixels.length; i++) {  int alpha = Color.alpha(pixels[i]);  int blue = Color.blue(pixels[i]);  newPixels[i] = Color.argb(alpha, 0, 0, blue);  }  return newPixels;  }  } |

|  |
| --- |
| ImageOverlay.java |
| package com.example.lab5;  import androidx.appcompat.app.AppCompatActivity;  import android.annotation.SuppressLint;  import android.graphics.Bitmap;  import android.graphics.BitmapFactory;  import android.graphics.Color;  import android.os.Bundle;  import android.view.View;  import android.widget.Button;  import android.widget.ImageView;  import android.widget.TextView;  public class ImageOverlayActivity extends AppCompatActivity {  ImageView original1, modified, original2;  Bitmap originalMap1, modifiedMap, originalMap2;  Button increaseAlpha, decreaseAlpha;  TextView alphaView;  double alpha = 0.7;  double step = 0.05;  int width = 0, height = 0;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_image\_overlay);  initViews();  initMaps();  initButtonListeners();  setModifiedImage();  }  private void initViews() {  original1 = findViewById(R.id.original1);  original2 = findViewById(R.id.original2);  modified = findViewById(R.id.modified);  increaseAlpha = findViewById(R.id.increase\_alpha);  decreaseAlpha = findViewById(R.id.decrease\_alpha);  alphaView = findViewById(R.id.alpha);  setAlphaText();  }  private void initMaps() {  originalMap1 = BitmapFactory.decodeResource(getResources(), R.drawable.overlay\_image\_1);  originalMap2 = BitmapFactory.decodeResource(getResources(), R.drawable.overlay\_image\_2);  original1.setImageBitmap(originalMap1);  original2.setImageBitmap(originalMap2);  width = originalMap1.getWidth();  height = originalMap2.getHeight();  modifiedMap = Bitmap.createBitmap(width, height, Bitmap.Config.ARGB\_8888);  }  private void initButtonListeners() {  increaseAlpha.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  increaseAlpha();  setModifiedImage();  setAlphaText();  }  });  decreaseAlpha.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  decreaseAlpha();  setModifiedImage();  setAlphaText();  }  });  }  @SuppressLint("DefaultLocale")  private void setAlphaText() {  alphaView.setText(String.format("%.2f", alpha));  }  private void decreaseAlpha() {  alpha -= step;  alpha = Math.max(0, alpha);  }  private void increaseAlpha() {  alpha += step;  alpha = Math.min(1, alpha);  }  private int[] overlayTwoImages() {  int[] pixels1 = new int[width \* height], pixels2 = new int[width \* height], newPixels = new int[width \* height];  originalMap1.getPixels(pixels1, 0, width, 0, 0, width, height);  originalMap2.getPixels(pixels2, 0, width, 0, 0, width, height);  for (int i = 0; i < pixels1.length; i++) {  int a = getOverlayedPixel(Color.alpha(pixels1[i]), Color.alpha(pixels2[i]));  int r = getOverlayedPixel(Color.red(pixels1[i]), Color.red(pixels2[i]));  int g = getOverlayedPixel(Color.green(pixels1[i]), Color.green(pixels2[i]));  int b = getOverlayedPixel(Color.blue(pixels1[i]), Color.blue(pixels2[i]));  newPixels[i] = Color.argb(a, r, g, b);  }  return newPixels;  }  private int getOverlayedPixel(int p1, int p2) {  return (int)(alpha \* (double)p1 + (1 - alpha) \* (double)p2);  }  private void setModifiedImage() {  int[] pixels = overlayTwoImages();  modifiedMap.setPixels(pixels, 0, width, 0,0, width, height);  modified.setImageBitmap(modifiedMap);  }  } |

|  |
| --- |
| FiltersActivity.java |
| package com.example.lab5;  import androidx.appcompat.app.AppCompatActivity;  import android.graphics.Bitmap;  import android.graphics.Color;  import android.os.Bundle;  import android.view.View;  import android.widget.Button;  import android.widget.ImageView;  import java.util.Arrays;  public class FiltersActivity extends AppCompatActivity {  ImageView view;  Button original, blur, sharpness, median, growingUp, sobel;  int w = MainActivity.mapWidth, h = MainActivity.mapHeight;  int [][] pixels2d;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_filters);  initViewsAndButtons();  original.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  view.setImageBitmap(MainActivity.originalMap);  }  });  blur.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  setUpOriginalMapTo2DPixels();  double [][] matrix = new double[][]{  {0.000789, 0.006581, 0.013347, 0.006581, 0.000789},  {0.006581, 0.054901, 0.111345, 0.054901, 0.006581},  {0.013347, 0.111345, 0.225821, 0.111345, 0.013347},  {0.006581, 0.054901, 0.111345, 0.054901, 0.006581},  {0.000789, 0.006581, 0.013347, 0.006581, 0.000789},  };  applyMatrix(matrix, 1);  }  });  sharpness.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  setUpOriginalMapTo2DPixels();  double [][] matrix = new double[][]{  {-1, -1, -1},  {-1, 9, -1},  {-1, -1, -1},  };  applyMatrix(matrix, 1);  }  });  median.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  setUpOriginalMapTo2DPixels();  applyMedianFilter();  }  });  growingUp.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  setUpOriginalMapTo2DPixels();  double [][] matrix = new double[][]{  {0, 0, 0.1, 0, 0},  {0, 0.1, 0.1, 0.1, 0},  {0.1, 0.1, 0.1, 0.1, 0.1},  {0, 0.1, 0.1, 0.1, 0},  {0, 0, 0.1, 0, 0},  };  applyMatrix(matrix, 1);  }  });  sobel.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  setUpOriginalMapTo2DPixels();  double[][] matrix = new double[][] {  { -1, -2, -1 },  { 0, 0, 0 },  { 1, 2, 1 },  };  applySobelFilter(matrix);  }  });  }  private void setUpOriginalMapTo2DPixels() {  int [] pixels = new int[w \* h];  MainActivity.originalMap.getPixels(pixels, 0, w, 0,0, w, h);  pixels2d = pixelsToTwoDimensionalArray(pixels, w);  }  private void initViewsAndButtons() {  view = findViewById(R.id.view);  original = findViewById(R.id.original);  blur = findViewById(R.id.blur);  sharpness = findViewById(R.id.sharpness);  median = findViewById(R.id.median);  growingUp = findViewById(R.id.growingUp);  sobel = findViewById(R.id.sobel);  view.setImageBitmap(MainActivity.originalMap);  }  private void applyMatrix(double [][] matrix, int div) {  final int subArrSize = matrix.length;  int level = subArrSize / 2;  for (int i = level; i < pixels2d.length - level; i++) {  for (int j = level; j < pixels2d[i].length - level; j++) {  int [][] subArr = getSubArr(i, j, subArrSize);  pixels2d[i][j] = multiplyMatrix(subArr, matrix, div);  }  }  setMapFromPixels2D();  }  private void applyMedianFilter() {  final int subArrSize = 21;  int level = subArrSize / 2;  for (int i = level; i < pixels2d.length - level; i++) {  for (int j = level; j < pixels2d[i].length - level; j++) {  int [][] subArr = getSubArr(i, j, subArrSize);  pixels2d[i][j] = getMedianColor(subArr);  }  }  setMapFromPixels2D();  }  private int getMedianColor(int[][] pixels2d) {  int red = 0;  int green = 0;  int blue = 0;  int[] pixels = twoDimensionalArrayToPixels(pixels2d);  int[] tmp = new int[pixels.length];  for (int i = 0; i < 3; i++) {  for (int j = 0; j < pixels.length; j++) {  if (i == 0) tmp[j] = Color.red(pixels[j]);  else if (i == 1) tmp[j] = Color.green(pixels[j]);  else tmp[j] = Color.blue(pixels[j]);  }  Arrays.sort(tmp);  if (i == 0) red = tmp[(int)(pixels.length / 2)];  else if (i == 1) green = tmp[(int)(pixels.length / 2)];  else blue = tmp[(int)(pixels.length / 2)];  }  int center = pixels.length / 2;  return Color.argb(Color.alpha(pixels[center]), red, green, blue);  }  private void applySobelFilter(double[][] horMatrix) {  int size = horMatrix.length;  int level = size / 2;  double[][] verMatrix = new double[size][size];  for (int i = 0; i < size; i++) {  for (int j = 0; j < size; j++) {  verMatrix[i][j] = horMatrix[j][i];  }  }  Bitmap horMap = getBitmapFromMatrixApplying(horMatrix, 1);  Bitmap verMap = getBitmapFromMatrixApplying(horMatrix, 1);  int width = horMap.getWidth();  int height = horMap.getHeight();  Bitmap finalResult = Bitmap.createBitmap(width, height, Bitmap.Config.ARGB\_8888);  int[] horPixels = new int[width \* width];  int[] verPixels = new int[horPixels.length];  horMap.getPixels(horPixels, 0, width, 0, 0, width, height);  verMap.getPixels(verPixels, 0, width, 0, 0, width, height);  double alpha = 0.5;  for (int i = 0; i < horPixels.length; i++) {  int horPixel = horPixels[i];  int verPixel = verPixels[i];  int red = (int)(Color.red(horPixel) \* alpha + (1 - alpha) \* Color.red(verPixel));  int green = (int)(Color.green(horPixel) \* alpha + (1 - alpha) \* Color.green(verPixel));  int blue = (int)(Color.blue(horPixel) \* alpha + (1 - alpha) \* Color.blue(verPixel));  horPixels[i] = Color.argb(255, red, green, blue);  }  finalResult.setPixels(horPixels, 0, width, 0, 0, width, height);  view.setImageBitmap(finalResult);  }  private int multiplyMatrix(int [][] pixels, double [][] matrix, int div) {  int length = pixels.length;  int sumRed = 0;  int sumGreen = 0;  int sumBlue = 0;  for (int i = 0; i < length; i++) {  for (int j = 0; j < length; j++) {  sumRed += (int)(Color.red(pixels[i][j]) \* matrix[i][j]);  sumGreen += (int)(Color.green(pixels[i][j]) \* matrix[i][j]);  sumBlue += (int)(Color.blue(pixels[i][j]) \* matrix[i][j]);  }  }  sumRed /= div;  sumGreen /= div;  sumBlue /= div;  sumRed = Math.max(Math.min(255, sumRed), 0);  sumGreen = Math.max(Math.min(255, sumGreen), 0);  sumBlue = Math.max(Math.min(255, sumBlue), 0);  int center = (int)(pixels.length / 2);  return Color.argb(Color.alpha(pixels[center][center]), sumRed, sumGreen, sumBlue);  }  private int[][] getSubArr(int i, int j, int size) {  int level = size / 2;  int [][] subArr = new int[size][size];  for (int i\_ = i - level, k = 0; k < size; i\_++, k++) {  for (int j\_ = j - level, l = 0; l < size; j\_++, l++) {  subArr[k][l] = pixels2d[i\_][j\_];  }  }  return subArr;  }  private int[][] pixelsToTwoDimensionalArray(int [] pixels, int width) {  int height = (int)(pixels.length / width);  int [][] pixels2d = new int[height][width];  for (int i = 0; i < height; i++) {  for (int j = 0; j < width; j++) {  pixels2d[i][j] = pixels[i \* width + j];  }  }  return pixels2d;  }  private int[] twoDimensionalArrayToPixels(int [][] pixels2d) {  int width = pixels2d[0].length;  int height = pixels2d.length;  int [] pixels = new int[width \* height];  for (int i = 0; i < height; i++) {  for (int j = 0; j < width; j++) {  pixels[i \* width + j] = pixels2d[i][j];  }  }  return pixels;  }  private void setMapFromPixels2D() {  Bitmap map = Bitmap.createBitmap(w, h, MainActivity.originalMap.getConfig());  map.setPixels(twoDimensionalArrayToPixels(pixels2d), 0, w, 0,0, w, h);  view.setImageBitmap(map);  }  private Bitmap getBitmapFromMatrixApplying(double[][] matrix, int div) {  final int subArrSize = matrix.length;  int level = subArrSize / 2;  int[][] finalPixels = new int[h][w];  for (int i = level; i < pixels2d.length - level; i++) {  for (int j = level; j < pixels2d[i].length - level; j++) {  int [][] subArr = getSubArr(i, j, subArrSize);  finalPixels[i][j] = multiplyMatrix(subArr, matrix, div);  }  }  Bitmap map = Bitmap.createBitmap(w, h, Bitmap.Config.ARGB\_8888);  map.setPixels(twoDimensionalArrayToPixels(finalPixels), 0,w,0,0,w,h);  return map;  }  } |

**Результати роботи програми**

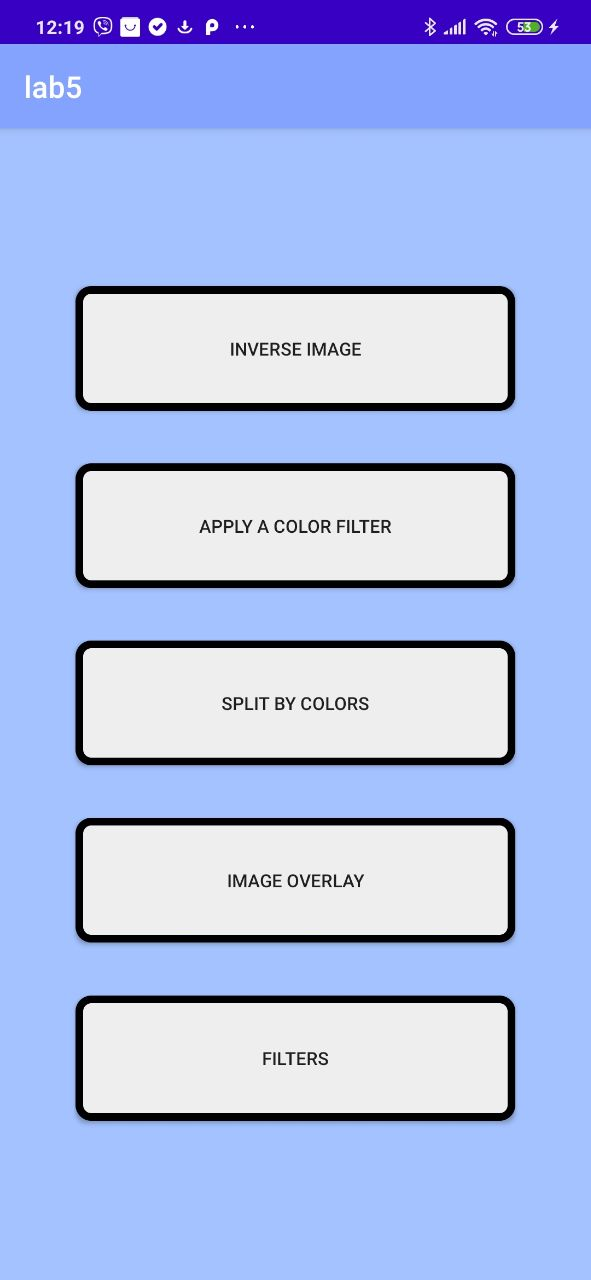


Рис 1.1, зовнішній вигляд головного меню

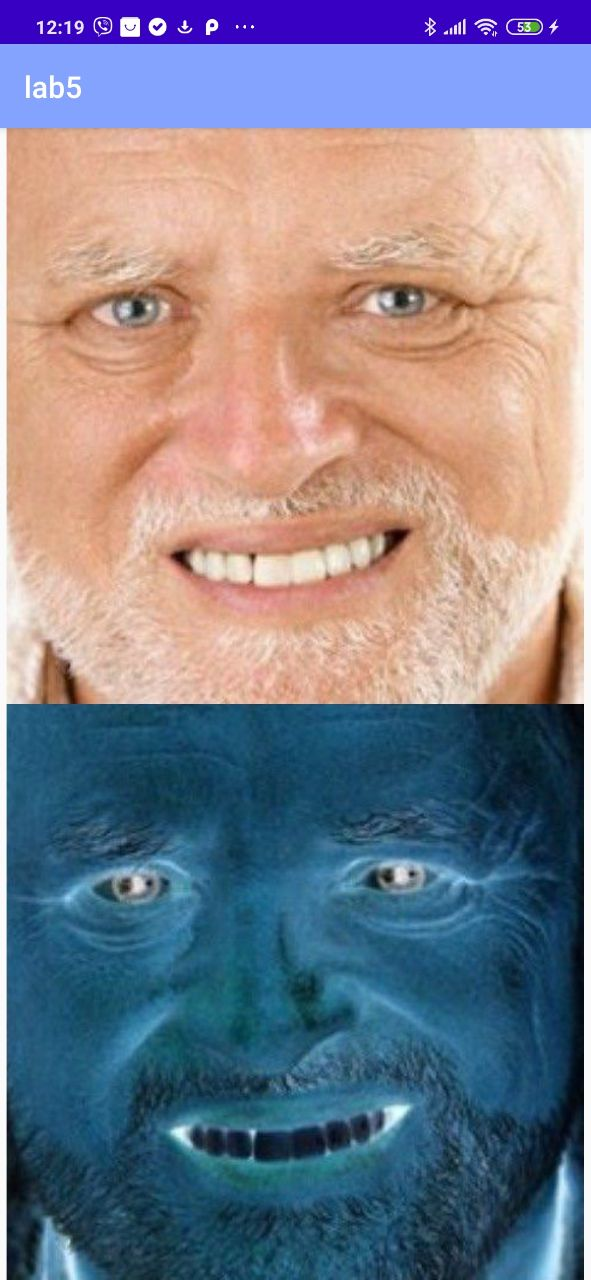


Рис 1.2, інверсія зображення



Рис 1.3, накладання кольорового фільтра

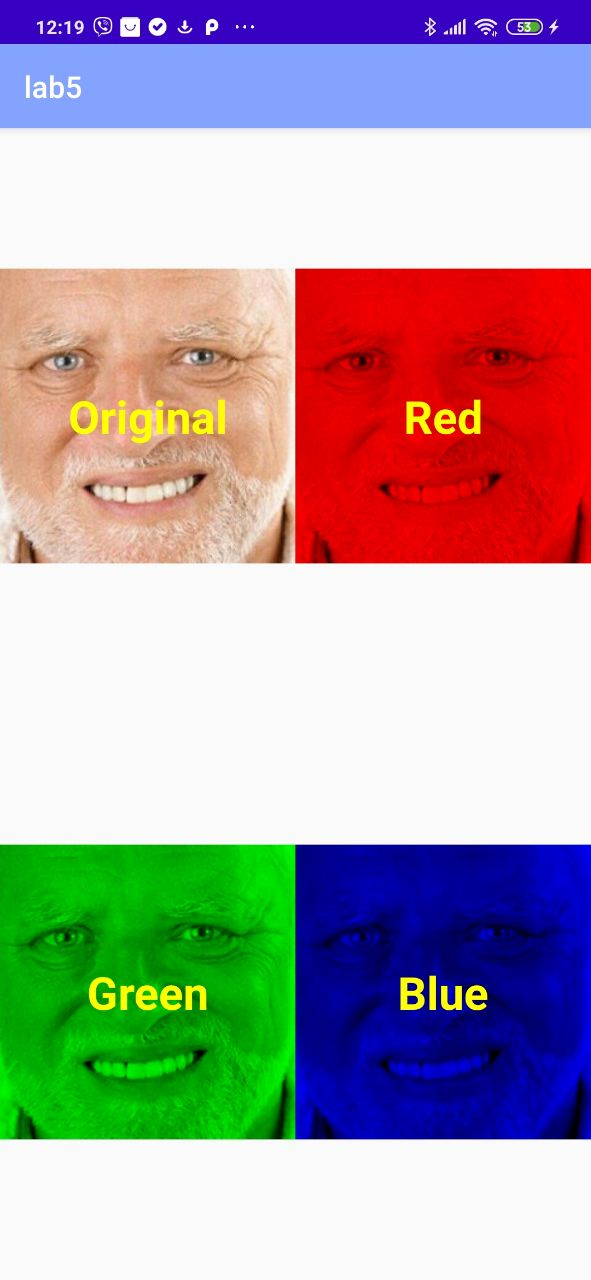


Рис 1.4, розбиття картинки по кольорам

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Рис 1.5.1, накладання картинок | Рис 1.5.2, накладання картинок | Рис 1.5.3, накладання картинок |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Рис 1.6.1, оригінал зображення | Рис 1.6.1, фільтр розмиття | Рис 1.6.1, фільтр підвищення чіткості |
|  |  |  |
| Рис 1.6.1, медіанний фільтр | Рис 1.6.1, зростання | Рис 1.6.1, фільтр Собеля |

**ВИСНОВКИ**

Під час виконання лабораторної роботи ми навчились із основними принципами обробки графічних даних з використанням основних методів і алгоритмів цифрової обробки зображень.