ASTM-FaceDetector

Studenți:

Anca Ioana Petre

Alexandru Sebastian Stoica

# Introducere teoretica

În sistemul de operare Android, o activitate reprezintă un ecran cu care utilizatorul poate interacționa pentru a face o acțiune, fiecărei activități corespunzându-i o fereastră. De fiecare dată când o nouă activitate începe, activitatea anterioară este oprită, dar păstrată într-o stivă numită „the back stack”. Trecerea de la o activitate la alta se face prin intermediul clasei Intent, care poate transmite parametri la activitatea următoare prin intermediul clasei Bundle.

Detecţia feţelor este tehnologia care se ocupă cu determinarea locaţiei şi dimensiunilor feţelor umane în diferite poze, ignorând orice altceva, cum ar fi clădiri, copaci sau corpuri. Detecţia feţelor poate fi privită ca un caz specific de detecţie obiect-clasă, ce se ocupă cu determinarea locaţiei şi dimensiunilor tuturor obiectelor ce aparţin unei clase într-o imagine dată.

Scopul acestei lucrări este de a integrara metoda Viola&Jones de detecție a feţelor. Metoda se bazaează pe detecția de trăsături de tip LPB calculate pe sub-regiuni din imagine folosind imaginea integrală, și identificarea prezenței feţelor umane cu ajutorul unui clasificator de tip cascadă. Abordarea propusă oferă capabilități de timp real și versatilitate pentru detecția oricaror tipuri de obiect pentru care s-a făcut o antrenare prealabilă a clasificatorului cascadă.

În OpenCV sunt disponibile modelele obținute prin antrenarea clasificatorului în format \*xml. pentru diverse tipuri de obiecte, dar lista nu este exhaustivă putând fi extinsă de către utilizatori.

Abordarea LBP folosește un descriptor de caracteristici binare, reprezentat ca un întreg fară semn. Acest lucru are ca rezultat:

* Timp de antrenare mai rapid și performanță de detectare a obiectelor, deoarece reprezentările binare ale caracteristicilor (LBP) sunt calculate și procesate mai rapid.
* O mai bună toleranță la variațiile iluminării comparativ cu abordarea Haar, deoarece pixelii sunt cuantificați la 0 sau 1 în funcție de "raportul" intensităților pixelilor și nu de intensitățile "efective" ale pixelilor.

Cu toate acestea, abordarea LBP aproximează caracteristicile (cuantificarea pixelilor) și face ca rezultatele obținute cu abordarea Haar să producă rezultate mai precise. Abordarea Viola-Jones oferă performanță în timp real și invarianță la scară / locație, dar are încă câteva dezavantaje - intoleranța la rotația obiectelor, sensibilitatea la variațiile de iluminare etc.

# Rezumatul funcționării aplicației

## Pentru detecţia feţelor am folosit algoritmul Viola – Johnes [1][2], implementat de bibliotecă open source OpenCV. Aceasta conţine deja clasificatorii antrenaţi, rezultatele antrenamentului fiind salvate în fişiere de tip XML. Pentru partea de antrenament a algoritmului, aceşti clasificatori trebuie încărcaţi în memorie.

Pentru partea de detectare a feţelor, pentru a fi posibilă detecţia în timp real, este nevoie de conectare la camera aparatului, fapt realizat prin intermediul clasei JavaCameraView, ce comunică cu componenta hardware a dispozitivului şi a interfeţei SurfaceView ce permite modificarea afişării pe ecran în mod programatic. La fiecare captură imaginea este preluată color, convertită în alb-negru şi este procesată, în următorii paşi:

* se defineşte o mărime minimă pentru încadrarea unei feţe (cu cât aceasta este mai mică, cu atât creşte complexitatea procesării imaginii);
* se iniţiază detecţia pe imaginea alb-negru apelând funcţia detectMultiScale() din clasa CascadeClassifier, salvându-se coordonatele feţelor detectate. - se desenează pătrate la coordonatele detectate pe imaginea color şi imaginea rezultată este afişată pe ecran.

## Explicarea principalelor clase/metode

## Principalele clase

Principalele clase ale proiectului sunt:

1. MainActivity.java
2. FaceDetector.java
3. sensorMotion.java

1. MainActivity este clasa principal care extinde AppCompatActivity oferind astfel o structură vizuală și elemente interactive care sunt familiare utilizatorilor. În clasa MainActivity s-au implementat mai multe variabile printre care imaginea în format rgba (mRgba), clasificatorul ViolaJones (mCascadeFile), detectorul de fețe din OpenCv (fd) și senzorul de rotație (sensorService).

Principalele metodele descrise în aceasta clasa sunt BaseLoaderCallback, onCreate, onCameraViewStarted, onCameraViewStop, onCameraFrame.

2. Facedetector este de asemenea una din clasele activității, clsasă ce definește un constructor ce instanțează automat cascada de clasificatori LBP și mai multe metode: setFrame, setRotetion, upScaleRectangle, detectFaces.

După ce se preia frame-ul de la MainActivity, se downsampleaza pentru viteza, este pasat detectorului și, după detecție, se stabilește rotația metoda upScaleRectagle ce upsamplează dreptunghiul de detecție de la rezoluția de lucru la vcea originală a frame-ului. Această metodă tratează toate cazurile de orientare.

În final metoda detectFaces este apelată iterativ din cameraFrame din Mainactivity.java pentru fiecare frame ce vine. Aici se convertește imaginea la gray, se rescalează la rezoluția de lucru și se aplica o corecție de contrast printr-o egalizare de histogframă. Pentru ficare range de orientare (cele 4 cadrane) se apelează funcția detectMultiScale pentru a detectea obiecte de dimensiuni diferite în imaginea de intrare, care apoi sunt returnate ca o listă de dreptunghiuri.

Detectia obiectelor se poate realiza apaland functia: void CascadeClassifier::detectMultiScale. CascadeClisifier – clasa din care se instantiata obiectul (modelul) pt. clasificatorul cascada. Modelul poate fi incarcat dintr-un fisier XML sau YAML folosinf functia Load().

După ce detecția a fost realizată, detectorul scoate afara un dreptunghi de încadrare. Acesta este scalat la dimensiunea normală a imaginii și translatat în funcție de orientarea la care s-a făcut detecția.

3. SensorMonitor este clasa standard ce extinde Activity pentru a implementează SensorEventListener și care se ocupă cu a updata/returna orientarea device-ului în timp real.

## Principalele metode

BaseLoaderCallback este o interfață de call back a instanțierii asincrone a obiectelor din OpenCV. Această interfață se ocupă cu încărcarea cascade de clasificatori LBP Viola Jones și creează obiectul de tip FaceDetector.

Metoda onCreate, onPouse, onResume, onDestroy instanțează clase java ce preiau bufferul video.

onCreate este locul unde vă inițializați activitatea. Cel mai important este că de obicei, se apelează setContentView cu un layout ce defineste interfața dorită și tot aici se folosește findViewById pentru a prelua widget-urile din interfața de utilizator.

onPause () este locul unde se tratează faptul că utilizatorul a părăsit activitatea. Chemată când sistemul urmează să înceapă să reia o activitate anterioară. Acest lucru este folosit de obicei pentru a comite modificări nesalvate ale datelor persistente, oprirea animațiilor și a altor lucruri care ar putea consuma CPU etc. Implementările acestei metode trebuie să fie foarte rapide deoarece următoarea activitate nu va fi reluată până când această metodă nu revine.



Fig.1 Activity lifecycle

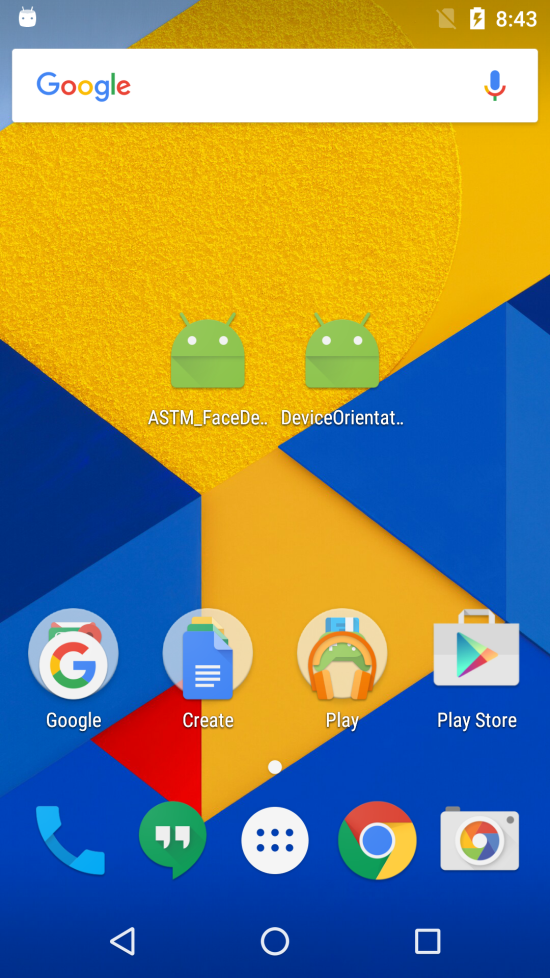
onResume este chemată când activitatea va începe să interacționeze cu utilizatorul. În acest moment, activitatea se află în partea superioară a stack-ului de activitate, intrarea venind de la utilizator.

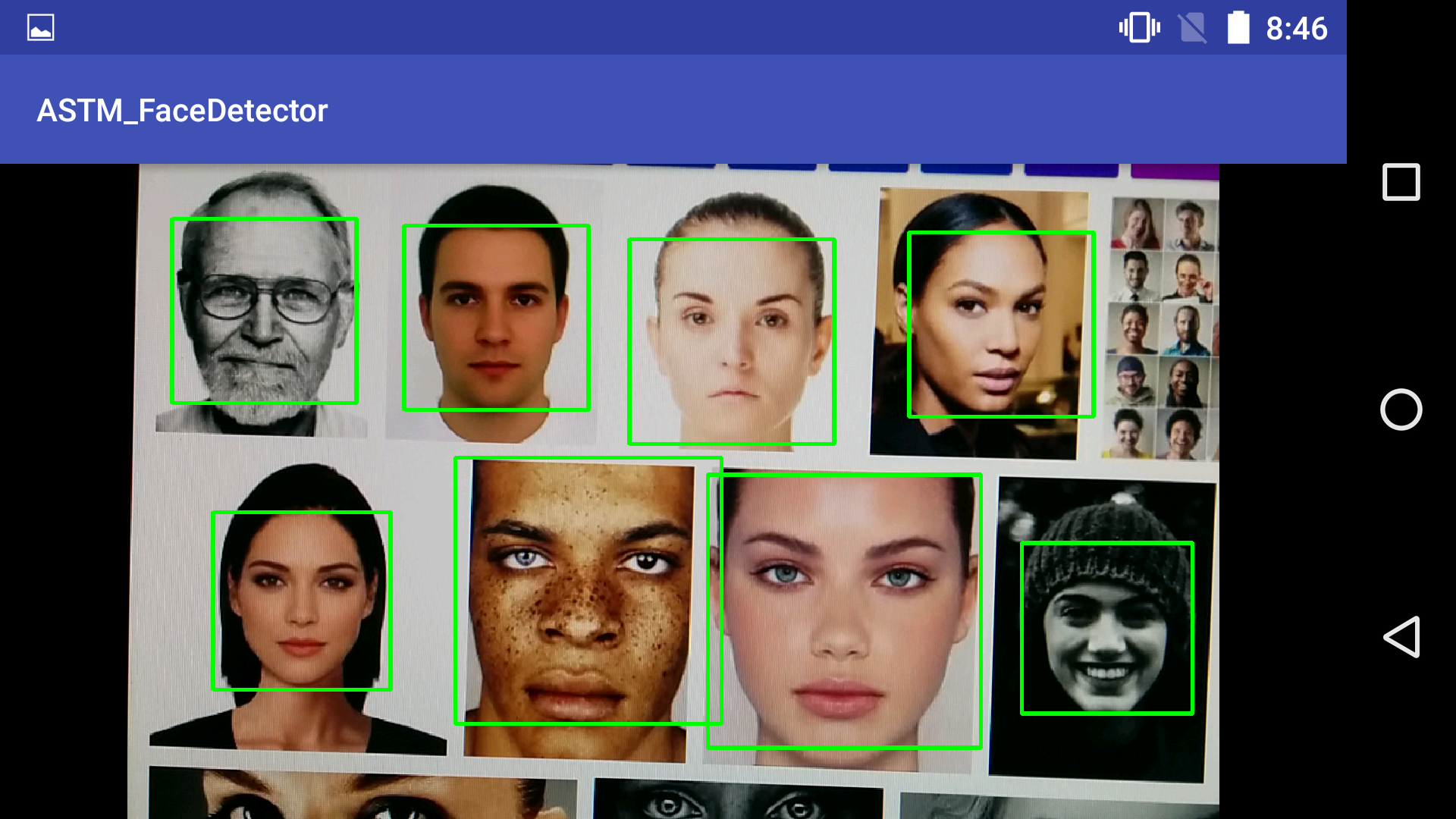
onDestroy este apelul final primit înainte ca activitatea sa fie distrus[. Acest lucru se poate întâmpla fie pentru că activitatea se termină, fie pentru că sistemul distruge temporar această instanță a activității pentru a economisi spațiu.

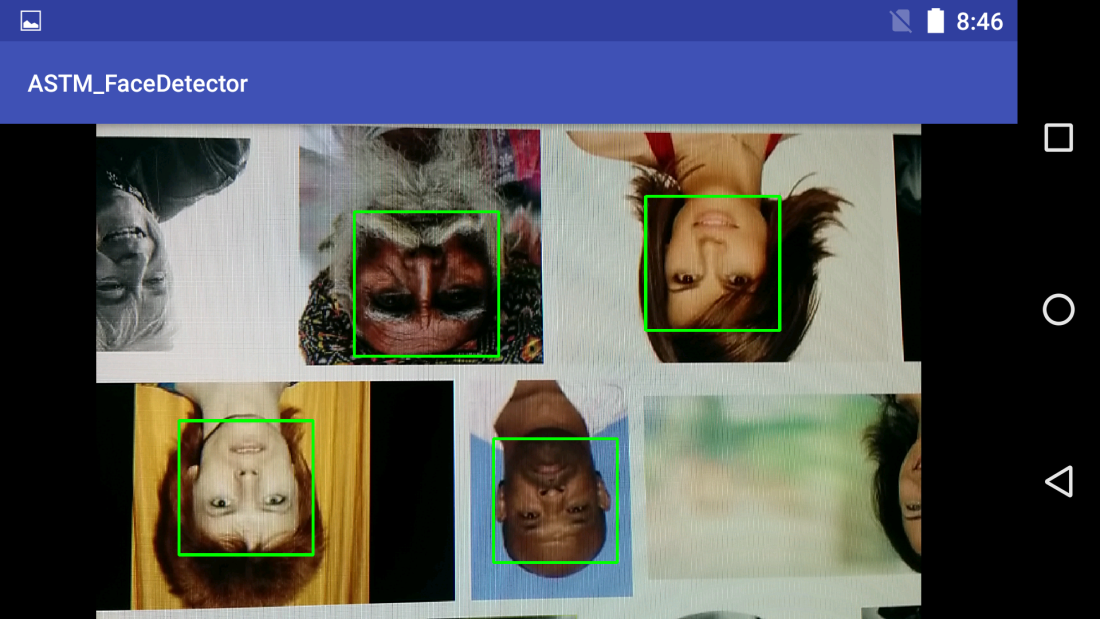
Metoda onCameraViewStarted se creează (alocă) un obiect de tip mat (OpenCV) de dimensiunea imaginii de intrare tip rgba pe 8biți.

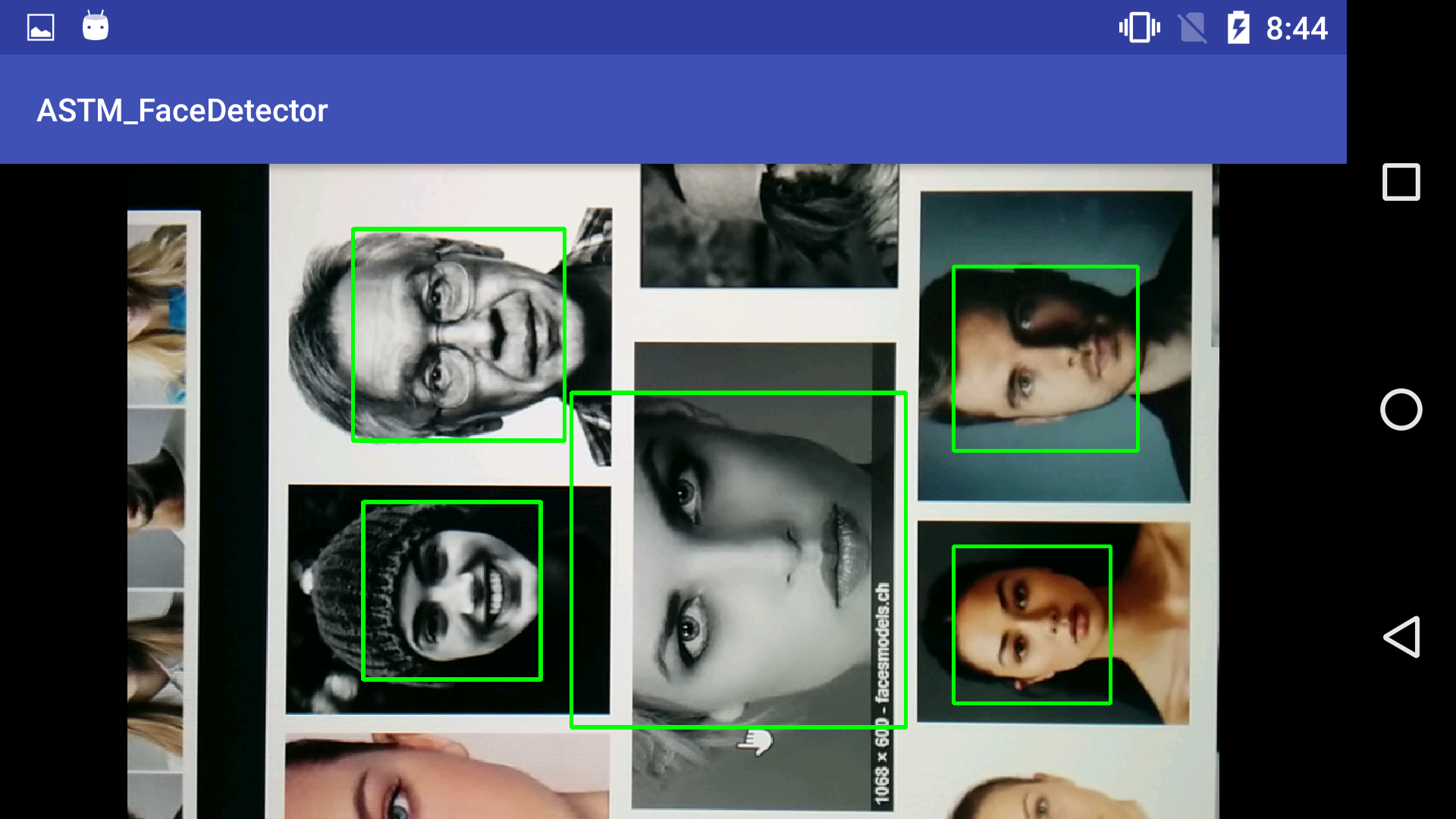
onCameraframe primește ca parametru frame-urile de la cameră, pe care le rotește în funcție de orientare și pe care le cheamă din funcția detectFaces dinFaceDetector.java, iar apoi returnează un obiect de tip mat.

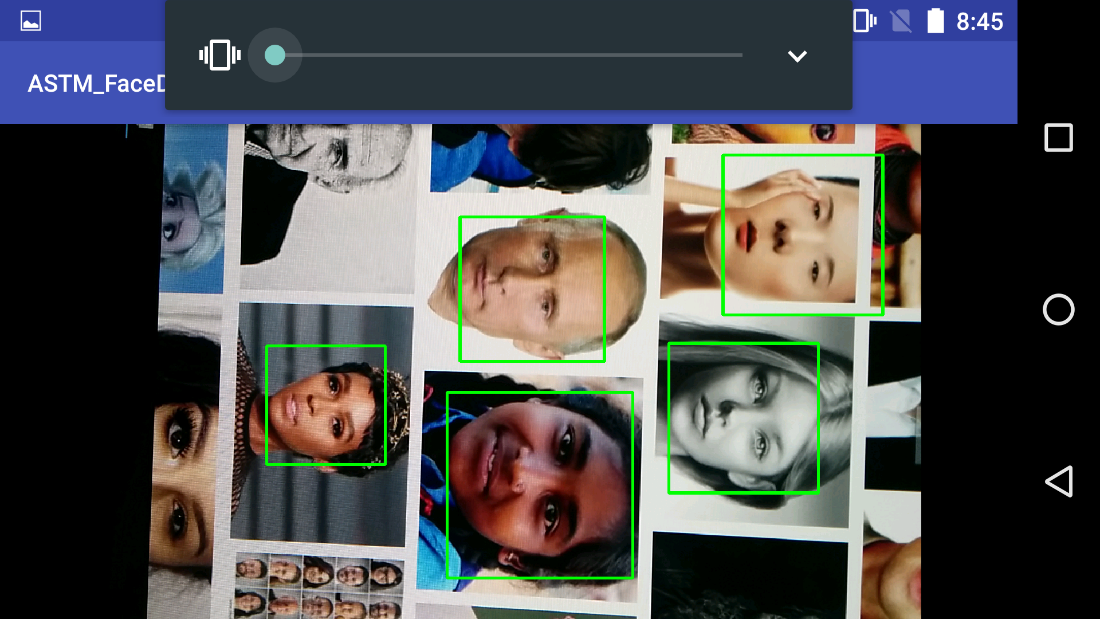
# Capturi de ecran – rulare aplicație diferite orientări











# Bibliografie

1. P. Viola, M. Jones: „Rapid Object Detection using a Boosted Cascade of Simple Features”, 2001;

2. P. Viola, M. Jones: „Robust real-time object detection”, 2001;

3. <http://opencv.willowgarage.com/wiki/FaceDetection>

4. OpenCV - <http://opencv.willowgarage.com/wiki/>