**Trandafir Irina, 1130 IE, Tehnici de procesare a imaginilor**

**Etapa I – Explicatii**

Tema de proiect pe care am ales-o pentru prima etapa este urmatoarea:

**3. Implementarea analizei PCA simplificate pentru un set de imagini.**

Datele de intrare necesare sunt:

A screenshot of a computer

Description automatically generated

**nrp** – care reprezinta numarul de poze

**tip** – care reprezinta extensia pozelor, tipul de fisier

**nrc** – numarul de componente principale pe care vrem sa il obtinem

Pozele utilizate in acest algoritm au denumirea urmatoare: **numarul pozei.png**. Dupa aplicarea acestui algoritm, se obtin pozele reconstituite folosind cele nrc componente principale. Pozele reconstituite au denumirea urmatoare: **numarul pozei\_r.png**.

Datele de iesire sunt:

**rez** - cod de terminare a operatiei (0=succes, 1=imagini cu dimensiuni diferite, 2=imagini cu mai mult de un plan)

**er** - eroarea (medie pe pixel)

**vect\_compresie** - componentele principale (nrc\*(m\*n)), vectorii proprii ai matricei de covarianta im\_lin\*im\_lin’

**medie** - vectorul (imaginea) medie ((m\*n)\*1)

Descrierea algoritmului utilizat in pseudocod:

A screenshot of a computer

Description automatically generated

Iau un contor k si cat timp contorul nu e egal cu numarul de imagini primit ca parametru si cat timp nu obtin un cod de eroare, execut while-ul. In while, creste contorul si construiesc numele imaginii de citit. Numele il construiesc folosind functia num2str (din numar transform in string) si cu ajutorul operatorului [ ] de concatenare lipesc de numarul pozei caracterul „.” si extensia primita ca parametru. Citesc imaginea si o stochez in variabila poza. Cu size stochez in m, n si p dimensiunile pozei. Verific daca poza are mai mult de 1 plan (nu e monocroma) si daca poza are mai mult de 1 plan, setez codul de eroare pe 2. Altfel, verific daca exista masivul tridimensional imagini si daca nu exista, aloc spatiu pentru acesta si setez variabilele m1 si n1 cu dimensiunea in pixeli a imaginii (pe m1 si n1 le voi folosi mai departe ca sa verific daca si celelalte imagini au aceleasi dimensiuni). Daca poza curenta are aceleasi dimensiuni ca prima poza citita, atunci o adaug in masivul imagini, altfel setez codul de eroare, rez pe 1.

A screenshot of a computer

Description automatically generated

Daca am obtinut o eroare si am iesit din while, k reprezinta numarul pozei la care am obtinut eroarea. Daca am o eroare afisez in linia de comanda mesajul „Imaginea k nu corespunde”) si in functie de eroarea obtinuta afisez mesajele „Are dimensiuni diferite” sau „Are mai mult de un plan”.

A screenshot of a computer

Description automatically generated

Daca nu am obtinut erori, atunci stochez in variabila p dimensiunea in pixeli a imaginilor, aloc spatiu pentru matricea im\_lin de dimensiuni p\*nrp in care voi stoca imaginile liniarizate pe coloane. Parcurg matricea im\_lin si pe coloane stochez fiecare imagine liniarizata. Pentru a liniariza imaginile, folosesc functia reshape. Reshape merge pe coloane, pentru a il face sa mearga pe linii, transpun imaginile. Furnizez si un size vector pentru a specifica dimensiunea imaginii liniarizate.

Calculez media coloanelor si o stochez intr-un vector de dimensiune [1 m\*n]. Scad din fiecare imagine liniarizata, vectorul medie, pentru a centra datele. Calculez matricea de covarianta astfel im\_lin’\*im\_lin. Calculez vectorii proprii ai matricei de covarianta si selectez acei vectori care corespund celor mai mari valori proprii ai matricei de covarianta. Din moment ce functia eig din matlab returneaza valorile proprii sortate crescator si vectorii proprii corespunzand fiecarei valori proprii si ei sunt in ordinea inversa fata de cea in care imi doresc, voi lua ultimii nrc vectori din matricea vectorilor proprii. Pentru a face asta merg de la finalul coloanelor, nrp pana la nrp-nrc+1 cu pasul -1.

A screenshot of a computer

Description automatically generated

Eroarea de reprezentare este egala cu suma valorilor proprii nefolosite. Vectorii proprii corespunzatori matricei cu imagini liniarizate inmultita cu matricea cu imagini liniarizate transpusa sunt vectorii proprii al aceluiasi produs de matrici scris invers, inmultiti cu matricea cu imagini liniarizate. Astfel pentru a obtine vectorii proprii ai matricei im\_lin\*im\_lin’ trebuie sa inmultim vectorii proprii ai matricei im\_lin’\*im\_lin cu im\_lin. Restul algoritmului decurge exact ca la KL, reprezentarea redusa se realizeaza schimband baza de reprezentare a imaginilor cu baza constituita din vectorii proprii ai matricei im\_lin\*im\_lin’, adica vectorii de compresie, vectorii proprii ai matricei im\_lin’\*im\_lin inmultiti cu im\_lin. Pentru a reconstitui imaginile trebuie sa revin la baza initiala. Tot ce a mai ramas de facut este sa adaug media si sa deliniarizez imaginile.

Exemple de apel:

[rez,~,~,er]=pca\_simplificata(400,'png',20);