

Dezvoltarea unei aplicații practice în cadrul unui proiect de semestru

~ Proiect ~

Intro Proiect

- **Proiectul își propune:**
 - să abordeze perspectiva practică din dezvoltarea unei aplicații complexe
 - ca o componentă a unui sistem ce include viziune computerizată.
- **Studentii se vor familiariza cu**
 - algoritmi de bază practici și simpli de prelucrare a semnalelor digitale,
 - oferind o posibilă bază de pregătire practică

pentru activități de dezvoltare și cercetare în acest domeniu.

Simplificări la nivel de disciplină

- Domeniul este foarte vast, astfel în cadrul orelor de la această disciplină se vor aplica câteva simplificări:
 - se discută doar *metode de baza*,
 - algoritmi mai avansați - presupun utilizarea cascadată/ combinată a metodelor de bază.
 - se tratează mai mult *imagini pe nivele de gri*;
 - generalizarea pentru imagini color fiind mult mai ușoară ulterior (și directă în unele cazuri)
 - se va realiza procesare pe *imagini mici* (de dimensiune/rezoluție redusă)
 - pentru a ne putea axa pe efectul procesării date de algoritmul nostru, fără a fi influențați de efectul dat de interpolare/zoom la faza de vizualizare/afișare rezultat prin redimensionarea imaginilor

Exemplu proiect: extragere muchii – în etapa de afișare a rezultatului se poate să fie eliminate părți din muchii prin procesul de sub-eșantionare a imaginii (pentru că nu vizualizăm complet rezultatul 1:1) - noi o să ne gândim că NU ne funcționează corect algoritmul – FALS, problema e la afișare.
 - se va lucra și cu *imagini de rezoluție mare, imagini color și secvențe video*
 - unde este relevant și necesar pentru temă/ algoritmul cerut (doar pe unele teme!)

Temele proiectelor urmăresc **structura cursului de la această disciplină** și se vor **realiza în echipe**.

De ce?

- De ce **bazele** si **NU notiuni avansate**?
- Care este **rolul** acestei discipline?

Proiect de semestru

- **Se dorește:**
 - înțelegerea principială și matematică sau logică a temei alese precum și:
 - proiectarea și implementarea practică - într-o aplicație software
- **Toate proiectele conțin**
 - parte de documentare,
 - implementare software și
 - validare experimentală,

având *aproximativ același grad de dificultate*.

Proiect de semestru

- **Nivele de intelegere/ dezvoltare a aplicatiei in cadrul proiectului de semestru:**

(1) Implementarea proiectului utilizând funcții gata implementate din pachetele publice online

- se vor folosi funcții gata implementate pentru partile de citire/ afisare/ operații intermediare

(2) Implementarea propriilor functii pentru partea matematica/ logica a algoritmilor din proiect

- se vor implementa propriile functii pentru partea de algoritmica din spatele temei
- legături între etape per proiect

Complexitate teme proiect

- Timpul de lucru și complexitatea pe proiect este similară pentru toate temele
 - chiar dacă în cazul unor proiecte:
 - *partea de implementare software pare mai ușoară*, partea de validare experimentală a algoritmilor este mai complexă, fiind necesare mai multe experimente cu diverși parametrii interpretați corect,
 - sau viceversa *partea software poate fi mai dificilă* dar validarea rezultatelor se realizează mult mai ușor.
- Toate temele sunt **interesante**, chiar dacă numele algoritmilor sunt de "neînțeles" momentan pentru voi!
- Proiectul trebuie să se finalizeze cu o aplicație software ușor de rulat de un utilizator nou.

Echipa

- **Prezența** la ședințele de proiect este **obligatorie**,
 - la fiecare ședință fiind necesară predarea unor livrabile, sau
 - prezentarea modului de lucru/ cum s-a avansat pe proiect
- **Proiectele se vor realiza în echipe de 2 studenți** (este recomandat ca cel puțin un membru din echipă să aibă cunoștințe de programare!)
 - existând activități de management, documentare, programare, testare și validare aplicație.
- **Rolurile in echipa se pot gândi astfel:**
 - **membrii echipei pot fi** - un manager și un dezvoltator.
 - Fiecare membru al echipei:
 - trebuie să cunoască și să se implice în sarcinile atribuite celui alt membru a echipei, pentru a se ajuta reciproc in a intelege/ implementa proiectul.

Echipa

- **Managerul** va trebui:
 - sa structureze optim si cat mai clar partea teoretica din spatele algoritmilor
 - să verifice/testeze fiecare pas din program cum că este corect realizat,
 - să realizeze raportul tehnic și un manual de utilizare a aplicației.

=> Activitățile de management: conduc la o înțelegere mai bună/extragere corectă și completă a părții teoretice, matematice sau logice a algoritmului.

- **Dezvoltatorul** va realiza aplicația software, fiind necesar sa se ocupe de:
 - implementare software si testarea algoritmilor din cadrul proiectului, conform studiului teoretic/raportului tehnic;
- **Etape** care se realizează de către **toți membrii echipei**:
 - documentarea teoretică,
 - instalare si setare mediu de programare si pachete conexe utilizate,
 - testare și validare aplicație.

Structura orelor de proiect

- **Descrierea modului de derulare a orelor de proiect, ETAPE:**

1) Formarea echipelor și alegerea unei teme de proiect;

2) Faza de documentare

- Prezentare PowerPoint pe tema proiectului (fiecare echipă)
(~6 slide-uri – extragere ecuații/parte logică ce urmează să fie implementate);

3) Implementare 10% din proiect – familiarizare cu mediul de programare.

- Testarea unor funcții de procesare din pachete disponibile gratuit online (pe direcția proiectului)
- Identificare date intrare/ date ieșire pentru proiect (de la ce plecăm/ ce trebuie să obținem)

4) Implementare între 30-80% din proiect

- Implementare pe baza teoriei structurate (partea matematică sau logică din spatele algoritmilor – nu utilizând funcțiile gata dezvoltate în pachete disponibile gratuit online)!
- Revizuire parte teoretică și includere în raportul tehnic a proiectului (vezi model);

5) Implementare 100%; revizuire/completare raport tehnic; validare rezultate experimentale de către manager; realizare prezentare.

6) Predarea și prezentarea proiectului

Predarea proiectului

- Pentru **predarea proiectului** se va realiza un folder, denumit după tema proiectului/ membrii echipei (sub forma unui cod), care va conține DOAR următoarele:
 - **Resurse aplicație software**
 - partea de implementare realizată (cod – toate fișierele necesare rulării; cod-ul trebuie sa fie ușor de rulat);
 - imagini pentru testarea programului (maxim 10).
 - **Raport tehnic** - realizat în Word (documentul se va preda DOAR în format electronic).
 - **Prezentare PowerPoint** - prezentare tehnică și comercială a „produsului software” realizat.
- Fiecare semi-grupa va primi o locație online (sub forma unui link) unde fiecare echipă va încarcă folderul cu resursele proiectului (încărcarea se va realiza înainte de a începe prezentările, acasă/sau în sala).
 - Va rog sa nu incarcati totul ca si o arhiva - sa putem sa deschidem direct prezentarea ppt din folder-ul incarcat! Se poate arhiva partea de cod/ imagini (daca este cazul), dar nu si ppt-ul!
- Fiecare student își va susține proiectul în fața colegilor; codul se va rula după prezentare.
 - realizați prezentarea astfel încât colegii voștri să înțeleagă proiectului vostru.
- **Observație!!!** Încărcarea online pe site-uri publice a oricărui tip de fișier realizat în cadrul orelor de proiect (raport tehnic, prezentare PowerPoint sau cod) este strict interzisă fără acordul coordonatorului de proiect.

Mod de notare

- Punctajul se va acorda ținând cont de: înțelegerea teoriei, implementare corecta (in ce grad), obținerea unor rezultate experimentale corecte, validarea corecta a algoritmilor (scenarii de testare adaptate metodelor/ algoritmilor implementați), organizare raport tehnic/ prezentare.
- Atribuire punctaj:
 - **5 puncte pentru modul de lucru în cadrul orelor de proiect** (pe parcursul semestrului), respectarea termenelor de realizare a activitatilor de la o sedinta de proiect la alta.
 - **5 puncte pentru predarea finala:**
 - 1 puncte pentru modul de realizare a raportului tehnic (documentația scrisă)
 - Selecția/structurarea informației relevante temei.
 - Includerea secțiunilor (capitolelor) cerute cu respectarea proporțiilor.
 - 2 puncte pentru modul de implementare a părții practice, precum si gradul de dificultate in abordarea problemei
 - 2 puncte pentru modul de realizare a prezentării proiectului
 - 0.5 puncte – exista o prezentare / si 0.5 puncte – exista un/o prezentator/oare
 - 0.5 puncte – prezentarea este bine structurata
 - 0.5 puncte – modul in care s-a răspuns la întrebări/ s-a înțeles algoritmul implementat

Nota obținută pe proiect reprezintă **25% din nota finală** la această disciplină.

Etape Proiect

Etapa I - Documentare

- **Obiectiv etapă de documentare:**
 - Studiu teoretic pentru identificarea părții matematice sau logice a algoritmilor de implementat (NU este necesar să înțelegem algoritmi – doar să-i identificăm) .
 - Realizarea listei bibliografice a documentelor consultate relevante temei proiectului
- Pentru ședința următoare de proiect se va pregăti:
 - o Prezentare PowerPoint (~6 slide-uri):
 - 1 slide – pagina de start: titlu proiect, membrii echipei proiect;
 - 1 slide – cerițe, date de intrare și date de ieșire
 - ~2-3 slide-uri - prezentarea algoritmului (parte matematică sau logică);
 - 1 slide – bibliografie
- Prezentarea va conduce la discuții/ observații benefice înțelegerii temei proiectului.
 - Urmăresc dacă au fost identificate corect ecuațiile sau partea logică care urmează să o implementați.
- Faza de documentare se realizează de toți membri din echipă – O SINGURĂ PREZENTARE!

Modalitatea de documentare

- Documentarea trebuie să se realizeze plecând de la:
 - *suportul pentru curs* de la această disciplină:
 - prezentările powerpoint și un set de cărți suport curs – disponibile online
 - *prin consultarea a altor cel puțin 5 surse bibliografice*
 - disponibile pe internet, în biblioteca Universității sau personală (studii online, articole științifice, cărți, capitole de cărți)
- Mare parte din documentele/ cărțile valoroase ca și conținut *sunt redactate în limba engleză*, astfel cel puțin jumătate din documentele consultate trebuie să fie în limba engleză.

Lista bibliografică

- Specificațiile tehnice, formule, tabele preluate din literatura de specialitate vor fi însoțite de referință bibliografică, specificată clar între paranteze drepte.
- **Lista bibliografică** - recomand: similar cu referințele din cărți
astfel, adresarea spre referință se face prin concatenarea numelui primului autor cu anul de apariție a documentului/ articolului/ cărții [**AutorAn**], deci nu prin numere.
 - acest mod ușurează păstrarea corespondenței corecte: idee – lucrare de unde a fost preluată ideea (prin adresarea cu un număr este mult mai ușor să ne încurcăm).
 - în cadrul lucrării încă nu știm ordinea în care o să rămână informația din studiile găsite - e mai ușor de mutat dacă referința nu este numerică (pentru referința numerică - primul articol referit trebuie să fie 1, dacă modificăm ordinea informației/ includem informație nouă - trebuie renumerotate referințele să fie referite în ordine crescătoare)
 - de asemenea, putem salva și local articolele cu includerea acestui ‘cod’ în numele fișierului.
- Nu se permite preluarea identică a textelor din documentele consultate.
 - referință spre un document = în acel document se găsesc detalii despre ideea introdusă în paragraful referit (NU înseamnă că ideea a fost copiată din acel document!). Astfel, spre un paragraf se pot pune mai multe referințe - spre toate documentele care oferă detalii despre ideea/ conceptul prezentat.
- Ecuațiile, tabelele și figurile se numerotează. Tabelele și figurile vor avea și titlu. Zonele de cod (chiar dacă sunt preluate ca și print screen/ imagine NU se numerotează și NU au titlu.
- Abrevierile și notațiile folosite vor fi explicate în clar.

Lista bibliografică – continuare

- **Lista bibliografică** trebuie să includă *datele de identificare a referinței complete*,
 - în funcție de tipul referinței: **carte** [Lima2019], **articol de conferință** [Li1996], **articol de revistă** [Siegwart2001], **capitol de carte** [Arai1999] sau este vorba de informații preluate de pe o **adresă de internet** [Wolfram].
 - [Lima2019] Author(s) (ex: P. Lima, A. Bonarini, M. Mataric), Name of Book in Italics, Publisher, Place of Publication (City, Country), year (ex. 2019).
 - [Li2010] Author(s), Title of conference paper, Proceedings of xxx xxx, Publisher, conference location (City, Country), month and year, pp. (page numbers: first-last).
 - [Siegwart2001] Author(s), Name of paper. Name of Journal in Italics, Vol., month and year of the edition, pp. (page numbers: first-last).
 - [Arai1999] Author(s), Name of paper, In: Name of Book in Italics, Name(s) of Editor(s), (Ed.), pp. (page numbers: first-last), Publisher, Place of publication (City, Country), Year.
 - [Wolfram] The Wolfram Solution for Image Processing, [Online 25.09.2022], <http://www.wolfram.com/solutions/industry/image-processing/>

Etapa II - Implementare 10%

- **Obiectiv etapă:**
 - Acomodare cu mediul de programare Python pentru procesarea semnalului/ imaginilor.
 - Implementare proiect utilizând pachete disponibile gratuit online.
 - Încercăm să înțelegem cum continuăm cu partea de implementare.
 - Implementarea se va realiza pe celule, în stilul lucrărilor de laborator, astfel să putem urmări mai bine pașii din implementare, precum și rezultate intermediare.
- **Mod de lucru** - scripturi notebook .jpynb, online sau local (pentru evidențierea pașilor implementați)
 - **Google Colab și Drive**, cu partajare conținut între membrii echipei.
 - Se definește modul de lucru utilizând Google Colab corelat cu Drive.
 - Se vor încărca în Drive: imagini pentru testare, prezentarea .ppt , precum și restul documentelor necesare dezvoltării proiectului. Partajăm informația cu partenerul de echipă!
 - **Instalarea și rularea locală** pe sistemul propriu de operare este recomandată, dar opțională în această etapă (fiind necesară până la sfârșit în ultima etapă).
- **Implementare utilizând pachete disponibile gratuit online**
 - Unele proiecte se pot realiza ușor cu funcții din pachete publice, dar la unele proiecte este mai complicat acest aspect - deci nu o să se poată să realizați proiectul utilizând funcții din pachete publice.
 - Astfel, doresc să încercați să folosiți funcții din pachete publice online, fie pentru a rezolva cerințele proiectului, fie pentru a realiza cerințe conexe proiectului sau pași intermediari.
- **Echipa:** Momentan putem avea teste diferite, scripturi separate - să se familiarizeze cu Python-ul fiecare membru. Ulterior decideți un mod de implementare comun pe care doriți să mergeți.
 - **Se implică toți membri ai echipei în realizarea acestei etape!**

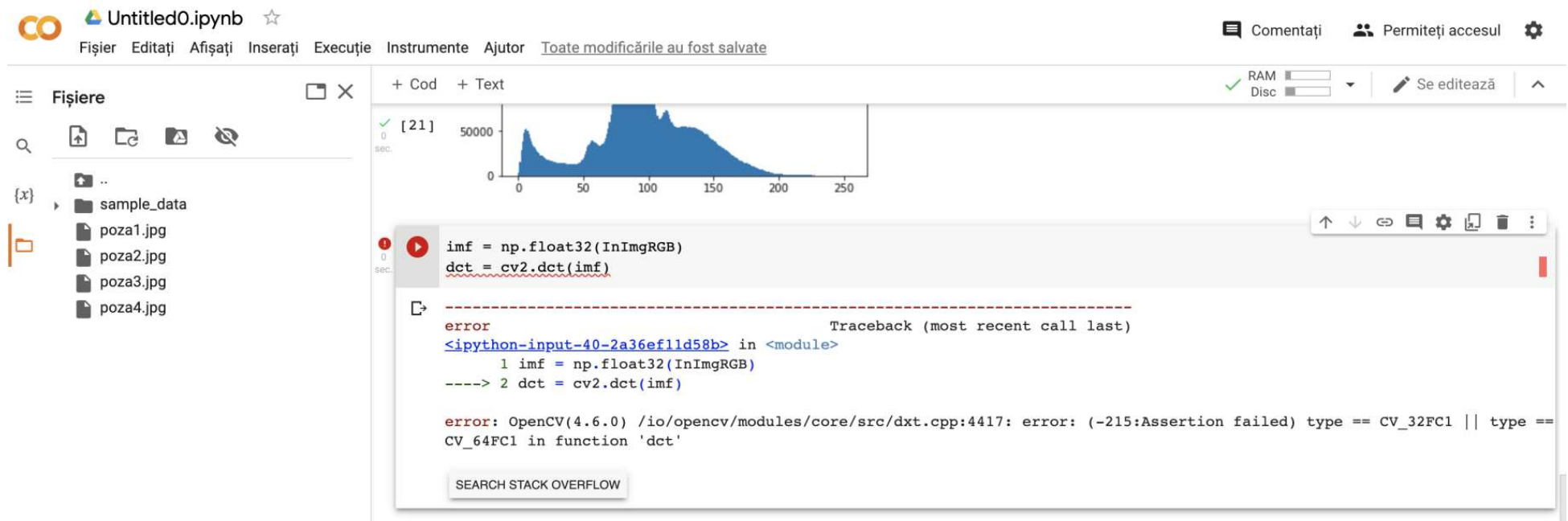
Probleme uzuale care apar

- Modalitatea de citirea a imaginilor difera in functie de biblioteca care o folosim
 - OpenCv-ul citeste imaginile ca si BGR si daca vrem sa le afisam cu Matplotlib avem nevoie ca ordinea canalelor sa fie RGB
- In Google Colab nu putem folosi functia `cv2.imshow(...)` pentru afisare imagine
 - Aceasta functie genereaza o fereastră externa si in Colab – lucrând in cloud, nu se poate genera aceasta fereastră
 - Exista versiune de afisare cu OpenCv direct pentru Colab

Probleme uzuale care apar

- Problemele uzuale care apar - mesaj primit de la o echipă pentru această etapă:

Buna ziua! Am incercat sa implementam etapa 3 de proiect in Google Colab folosind functiile oferite de opencv, dar am intalnit o eroare si nu putem sa folosim functia .dct pentru ca nu o recunoaste. Aveti idee cum am putea rezolva asta?



- Greșeala e că se aplică funcția pe imaginea color, funcția DCT are nevoie să aibă la intrare o matrice 2D nu o intrare cu 3 dimensiuni (cum e imaginea color).
- Trebuie să înțelegem cum să aplicăm corect algoritmi pe imagine, și nu uităm – majoritatea algoritmilor se aplică la nivel de canale sau pe versiunea imaginii pe nivele de gri!

Etapa III - Implementare 30-80%

- **Obiectiv etapă:**
 - ***Implementare pe baza teoriei structurate***
implementare partea matematică sau logică din spatele algoritmilor - aplicare pe imagine = o matrice 2D, NU utilizând funcțiile gata dezvoltate în pachete disponibile gratuit online!
 - rămâne și implementarea cu funcții din pachete publice (NU le ștergem)
 - ***Revizuire parte teoretică și includere în raportul tehnic a proiectului*** (vezi model "*RaportTehnic.docx*")
 - se completează la partea de introducere/ fundamentare teoretică și parțial implementare.
- Fiecare membru de echipă ***va avea propriile atribuții***, dar ambii membri ai echipei trebuie să se gândească și să discute care este următorul pas în implementarea algoritmului echipei.
 - Un membru se concentrează pe implementare celălalt pe raportul tehnic.

Etapa III - Implementare 30-80%

- Observații în urma ședințelor de proiect anterioare:
 - Din fiecare echipa, un rol a managerului este să citească mesajele mele, pentru a fi la curent cu observațiile mele transmise spre voi.
 - Se va include în cod o celulă cu titlul proiectului/ cerințe proiect – să-mi fie clar despre ce discutăm când vorbim pe cod.
 - Imaginile vreau să fie citite din Google Drive – corelare Google Colab și Drive
 - Atenție la modul de deschidere a imaginii și utilizarea ei, reprezentarea corectă este: când folosim imagine pe nivele de gri ($H \times W$), când folosim imagine color ($H \times W \times 3$).
 - Uzul, pentru majoritatea proiectelor lucrăm pe imaginea pe nivele de gri.
 - Atenție, reprezentarea imaginii pe nivele de gri este ca o matrice de dimensiune ($H \times W$).
 - Dacă este o matrice de dimensiune ($H \times W \times 3$), dar la afișare arată ca o imagine pe nivele de gri, atunci cele trei canale sunt egale $R=G=B$ și avem reprezentarea RGB, nu Gray. Astfel, pentru citirea imaginii pe nivele de gri utilizând pachetul *OpenCv* se folosește corect setând la funcția de citire parametru al doilea egal cu 0 - *cv2.imread(numeImg, 0)*; sau utilizare conversie RGB spre Gray dacă imaginea este citită RGB.
 - La afișare imagini pe nivele de gri utilizând pachetul *pyplot* se specifică '*cmap=gray*'.
 - Unele proiecte necesită imagini color – fie pentru procesare, sau afișare a rezultatelor (ilustrare cu roșu peste imagine a rezultatelor), imaginea color se tratează pe 3 canale ($H \times W \times 3$).
 - Nu uităm, utilizăm imagini de rezoluție redusă, pentru a ne putea axa pe efectul procesării date de algoritmul nostru, fără a fi influențați de efectul de interpolare/ zoom de la vizualizare prin redimensionarea imaginilor.

Etapa IV - Implementare 99%

- **Obiectiv etapă:**
 - Finalizare implementare - rămâne să ne concentrăm pe partea de validare rezultate experimentare (testare algoritm)
 - selecție imagini de lucru, optimizare parametrii în funcție de imaginea aleasă (dacă este cazul), etc.
 - Raport tehnic – tot, excepție partea de validare a rezultatelor experimentale.

Etapa V - Predare Proiect

- Finalizarea tuturor task-urilor
 - Implementare finală și validarea rezultatelor experimentale
 - Rulare atât în Google Colab cât și local (fișier .ipynb, și .py).
 - Fișierele din Colab se pot descărca și cu extensia .py și rula în Spyder.
 - Raport tehnic final
 - Prezentare PowerPoint finală
- Pentru **predarea proiectului** se va realiza un folder (ne-arhivat), denumit după tema proiectului/ membrii echipei (sub forma unui cod), care va conține DOAR următoarele:
 - **Resurse aplicație software**
 - partea de implementare realizată (cod – toate fișierele necesare rulării; cod-ul trebuie să fie ușor de rulat) - versiunea din Colab, salvată direct din Colab ca și .py, și ca și pdf;
 - imagini pentru testarea programului (maxim 10).
 - **Raport tehnic** - realizat în Word (documentul se va preda DOAR în format electronic).
 - **Prezentare PowerPoint** - prezentare tehnică și comercială a „produsului software” realizat.

Prezentare PowerPoint Finala

- **Prezentare PowerPoint Finală** - prezentare tehnică și comercială a „produsului software” realizat care va conține:
 - sinteza temei de proiect;
 - descrierea teoretică a algoritmului implementat (puțin text; descriere matematică);
 - aspecte legate de partea de implementare (doar codul relevant pentru implementarea părții teoretice, asociate pașilor algoritmului implementat – fără toate funcțiile de afișare rezultate);
 - includerea și descrierea rezultatelor experimentale proprii (pe ~ 5 imagini) - cu axare pe rezultate corecte, când/ în ce condiții algoritmul funcționează optim;
 - concluzii/ perspective viitoare de continuare a temei;
 - lista bibliografică (și includerea referințelor bibliografice în textul prezentării).
- **Rularea codului după prezentarea ppt-ului proiectului** – după prezentarea tuturor proiectelor, pot cere deschiderea codului pentru a îmi reîmprospăta memoria cu toate părțile proiectului
 - Din partea de prezentare – trebuie să reiasă toate aspectele importante, și contribuțiile, pe părțile de:
 - teorie/ implementare / rezultate experimentale
 - Rularea codului după prezentare este doar să văd că este funcțional!!!! Nu o să puteți/ aveți timp să-mi explicați ce face/cum funcționează codul atunci – trebuie să reiasă din ppt!
 - Posibil să nu vă mai cer rularea codului, având în vedere că l-am văzut pe parcurs!