



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Identificarea si monitorizarea factorilor de risc interni si externi pentru siguranta traficului auto - INOVSAFE

Echipa de cercetare: Răzvan Itu, Radu Dănescu

ETAPA 1 - Achizitie date senzoriale, proiectare sistem si extragere date relevante

Cuprins

Introducere - rezumatul etapei	2
1. Achizitie date senzoriale	2
2. Proiectare, antrenare si evaluare retele neuronale si algoritmi pentru a extrage date relevante din traficul rutier	4
3. Proiectare, antrenare si evaluare retele neuronale si algoritmi pentru a monitoriza starea soferului si a vehiculului propriu	5
4. Implementarea sistemului de procesare pe sisteme desktop	7
Publicatii	8
Concluzii	8
Bibliografie	8

Introducere - rezumatul etapei

Etapa 1 a avut patru direcții majore:

1. Achiziție date senzoriale
2. Proiectare, antrenare și evaluare rețele neuronale și algoritmi pentru a extrage date relevante din traficul rutier
3. Proiectare, antrenare și evaluare rețele neuronale și algoritmi pentru a monitoriza starea soferului și a vehiculului propriu
4. Implementarea sistemului de procesare pe sisteme desktop

1. Achiziție date senzoriale

Datele senzoriale pot fi achiziționate folosind sisteme hardware proprii, compuse fie din aparate foto/video montate pe vehicule, sau folosind dispozitive mobile (ex. terminale mobile cu sistem de operare Android) dotate cu aparate foto/video integrate.

Pentru acest proiect, am utilizat un sistem de achiziție folosind dispozitive mobile portabile cu sistem de operare Android. Avantajul este dat de faptul că acest sistem este open source la baza, iar dezvoltarea de aplicații capabile să acceseze camera se poate implementa fără a fi necesară achiziția de licențe sau unelte de dezvoltare dedicate. Mai exact, achiziția de imagini se poate face prin bibliotecile software disponibile pe sistemul Android. Dezvoltarea unui astfel de software de achiziție se poate face fie prin unealta Android Studio sau prin IntelliJ.

Sistemul de achiziție propus în acest proiect va prelua imagini din camera din spate a dispozitivelor mobile, care este orientată spre carosabil (spre fața vehiculului propriu). Totodată, am propus achiziția și de date senzoriale inertiabile ale vehiculului propriu, care vor fi folosite pentru analiză și predicția unor situații periculoase din trafic.

Datele senzoriale sunt accesate folosind bibliotecile software disponibile și accesează informații direct din senzorii dispozitivelor mobile, cum ar fi: accelerometru, compas, senzori de poziționare prin satelit (GPS, GLONASS), orientare magnetică (magnetometru pentru orientare = busolă digitală), giroscop. Acești senzori pot fi folosiți pentru a efectua calibrarea dispozitivului mobil și pentru a determina matricea de transformare a coordonatelor din spațiul imaginii în spațiul lumii ("camera coordinates to world coordinates"). Accelerometrul și giroscopul oferă informații despre orientarea dispozitivului mobil în raport cu vehiculul și implicit cu lumea (scena de trafic). Senzorii de poziționare prin satelit oferă informații despre viteza de deplasare și poziționarea din orașe, intersecții sau autostrăzi pe care se circula. Prin fuziunea acestor informații se pot estima informații și despre restul participanților din trafic și astfel se pot determina situații periculoase.

Un alt sistem de achizitie folosit este cel disponibil in laboratorul de cercetare, compus dintr-un aparat foto digital de tip DSRL, cu lentila de tip “wide-angle”. Achizitia cu un astfel de sistem este controlata dintr-un sistem portabil (notebook). Captura imaginilor este realizata folosind un software propriu realizat in linia de comanda care va controla un microcontroller ce are rolul de a genera un semnal folosit de aparatul foto pentru declansare (similar cum sunt folosite telecomenzile pentru aparate foto DSRL). Imaginile sunt salvate pe cardul de memorie intern al aparatului foto si pot fi transferate ulterior procesului de achizitie, sau se pot copia automat prin folosirea unui cablu de date si a unui software scris in linia de comanda pentru transfer de fisiere. Procesul de achizitie este ilustrat in figura 1.

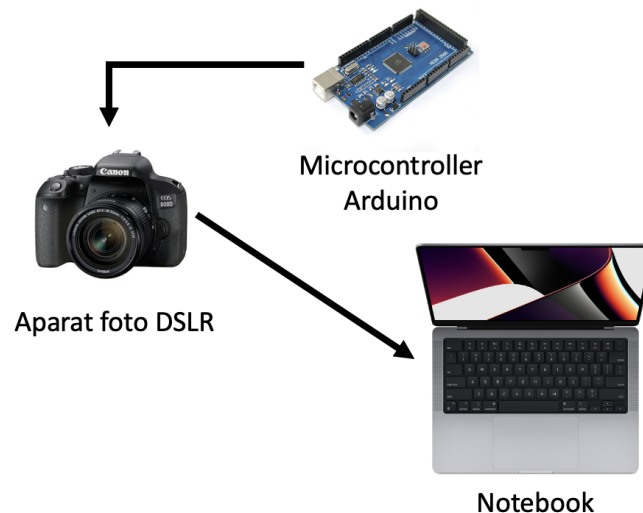


Figura 1. Procesul de achizitie a imaginilor de trafic folosind un microcontroller bazat pe Arduino pentru declansarea capturii de imagini la intervale regulate de timp cu un aparat foto DSLR si transmisia imaginilor pe un sistem portabil (notebook/laptop).

O alta varianta pentru achizitie de imagini (folosind hardware din laboratul de cercetare) este compusa dintr-un microcontroller de tip Raspberry Pi si o camera dedicata (PiCamera). Microcontroller-ul Raspberry Pi ofera suficienta putere de procesare pentru a captura imagini cu camera dedicata. Datele pot fi transferate ulterior pe un sistem desktop pentru a antrena o retea neuronală si pentru a putea fi interpretate si procesate.

Achizitia datelor se poate face si intr-un mediu controlat, din laboratorul de cercetare, folosind anumite simulatoare 3D. Pentru acest proiect am instalat si configurat simulatorul Carla [1]. Acest software este “open source” si este sustinut de lideri din industria automotiva si software: Toyota Research Institute, Nvidia, Intel. Simulatorul Carla este destinat domeniului vehiculelor autonome si ofera diferite scenarii din mediul urban sau de pe autostrada, din care se pot achizitiona imagini si date senzoriale (puncte 3D - “point cloud”, segmentare semantica, imagini de disparitate - “stereo disparity”, etc.). Software-ul permite controlul manual sau automat al unui vehicul care navigheaza prin diferite harti/scenarii.

2. Proiectare, antrenare si evaluare retele neuronale si algoritmi pentru a extrage date relevante din traficul rutier

Extragerea de trasaturi relevante din imaginile din traficul rutier se poate face prin retele neuronale convolutionale, capabile sa ofere predictii in timp real (procesarea unei imagini efectuata in sub 50 milisecunde, in functie de hardware-ul folosit). Cele mai utilizate metode pentru perceptia scenariilor de trafic rutier folosesc retele neuronale convolutionale cu o arhitectura (structura) de tip "encoder-decoder", in care imaginile de intrare ale scenei sunt trecute printr-o retea care extrage datele relevante (trasaturile) si apoi acestea sunt folosite intr-o retea care va genera o imagine cu rezultatele dorite (segmentare semantica, imagini cu harta de adancime - "depth map", flux optic - "optical flow").

Proiectul propus va folosi o retea de tip U-Net [2], care este bazata pe arhitectura "encoder-decoder" (codificare - decodificare) si este utilizata de obicei pentru analiza imaginilor medicale. Aceasta solutie se preteaza si pentru imagini din vehicule ale scenei de trafic, iar antrenarea retelei se poate face usor si in timp redus (numar de parametri antrenabili redus). Retelele bazate pe U-Net ofera predictii bune, chiar daca sunt antrenate cu putine imagini/date de intrare, deoarece U-Net foloseste conexiuni directe intre nivelurile de "encoder" si "decoder" (ceea ce ajuta la procesul de antrenare cu o cantitate redusa de date).

In implementarea proiectului am folosit o prima varianta scrisa in limbajul de programare Python pentru biblioteca PyTorch, iar ulterior am folosit biblioteca Keras.

Totodata, am testat si arhitectura ResNet [3], care a introdus conceptul de blocuri reziduale ("residual network") cu rolul de a evita problemele aparute din timpul antrenarii retelei. Se introduc astfel conexiuni directe (scurtaturi = "shortcuts") intre nivelurile modulului de codificare. Aceste legaturi sunt grupate in niveluri numite blocuri de identitate ("ID block"). Am folosit o varianta modificata, numita ResNet50 care are 48 niveluri de convolutie, 2 niveluri de agregare (1 nivel MaxPool si 1 nivel AveragePool). Varianta proprie a implementarii folosind ResNet50 ca si modul de codificare ("encoder") si structura similara cu U-Net pentru modulul de decodificare ("decoder") este ilustrata in figura 2 (unde "bloc ID" se refera la "ID block"). In structura retelei, nivelul "decodare" se refera la operatia de convolutie transpusa (numita si deconvolutie sau "upsampling") si are rolul de a inlocui operatiunile traditionale de interpolare biliniara pentru a construi harta (imaginea) de segmentare semantica a scenei la dimensiunea imaginii de intrare.

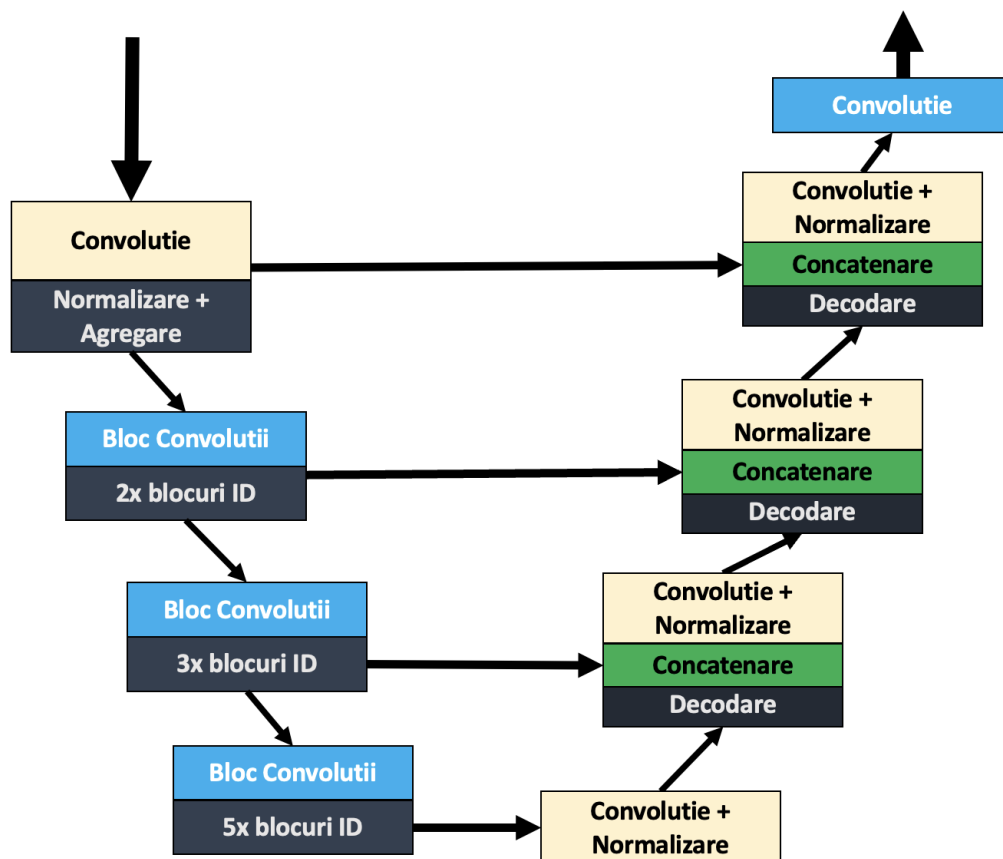


Figura 2. Structura rețelei bazată pe ResNet50 bazată pe arhitectura de tip “encoder-decoder”.

Această arhitectură de rețea neuronală convoluțională a stat la baza percepției traficului cu scopul de a identifica obstacolele prezente în fața vehiculului propriu și a fost valorificată prin publicarea într-un jurnal indexat ISI [4].

3. Proiectare, antrenare și evaluare rețele neuronale și algoritmi pentru a monitoriza starea soferului și a vehiculului propriu

Analiza temporală a unei secvențe de imagini reprezintă un domeniu activ de cercetare și are rolul de a oferi predicții mai bune asupra evenimentelor dintr-o scenă de trafic rutier (în cazul acestui proiect de cercetare). Primele lucrări în acest sens au vizat crearea unei rețele neuronale capabile să genereze comenzi pentru unghiul de virare (“steering angle”) al unui vehicul, direct din imaginea de intrare [5]. Alte lucrări au propus analiza traiectoriei pietonilor [6], [7] sau a vehiculului propriu [8]. Aceste abordări sunt de tipul “end-to-end” și au rolul de a oferi predicții asupra unor stări curente sau viitoare ale vehiculului sau acțiuni care sunt necesare (menținere bandă curentă, virare la stanga, etc.). Tendința este de a învăța din bazele de date existente modele de conducere automată care pot fi aplicabile direct unui

vehicul sau robot autonom. O lucrare de referinta este [9], in care autorii au folosit imagini si date sintetice din simulatorul Carla pentru a oferi predictii asupra unor actiuni care pot fi aplicate unui vehicul. Astfel, problema este tratata ca una de inteligenta artificiala in care un agent este "invatat" sa dea comenzi/actiuni unui vehicul pentru a naviga in siguranta in anumite scenarii. Sistemul din lucrarea mentionata anterior a fost testat folosind acelasi simulator. Solutia are la baza folosirea unor imagini a scenei de trafic impreuna cu un vector de masuratori si a unei comenzi pentru a prezice comanda/actiunea urmatoare care trebuie aplicata. Aceasta abordare necesita crearea unei baze de date in care pentru fiecare imagine de intrare sa fie disponibila si acea comanda/actiune, ceea ce poate duce la un efort suplimentar in procesul de antrenare a rețelei neuronale.

Starea soferului si a vehiculului propriu se poate determina prin analiza unei secvente de imagini si prin analiza informatiilor achizitionate de la senzorii vehiculului (informatii despre viteza de deplasare, pozitie, acceleratie, orientare: unghiul de rotatie ("roll"), inclinare ("pitch"), unghiul de giratie - "yaw", etc.). Fuziunea acestor date impreuna cu informatiile referitoare la perceptia scenei sunt esentiale pentru invatarea unui model capabil sa controleze un vehicul sau robot autonom.

Sisteme de predictie a pietonilor sau vehiculelor din contexte de trafic rutier au fost propuse in ultimii ani folosind rețele neuronale recurente ("RNN - recurrent neural networks"). Acestea au anumite dezavantaje in momentul antrenarii rețelei (problema actualizarii gradientilor din timpul invatarii/antrenarii din cauza structurii rețelei si a functiilor de activare folosite - "vanishing gradient problem" [10]), iar limitarile au fost rezolvate prin folosirea rețelelor de tip "LSTM" ("Long Short Term Memory"). Acestea au o structura formata din 3 parti: "input gate", "forget gate" si "output gate" [11].

Pentru acest proiect, am urmarit implementarea unei arhitecturi de retea convolutionala capabila sa extraga anumite trasaturi comune din imaginea de trafic rutier, care sa combine si informatiile senzoriale si sa genereze predictii asupra scenei si a starii vehiculului curent. Astfel, am propus utilizarea unei rețele de tip "encoder" pentru extragerea trasaturilor relevante si doua module: unul de decodificare ("decoder") pentru segmentarea semantica si analiza imaginii scenei, si un modul care sa utilizeze datele senzoriale folosind o structura de retea de tip LSTM. Predictiile din aceste doua module sunt concatenate si vor fi folosite apoi de alta retea LSTM pentru predictia finala. Structura propusa a rețelei este ilustrata in figura 3 si are la baza sistemul de predictie a traiectoriei pietonilor din lucrarea [12].

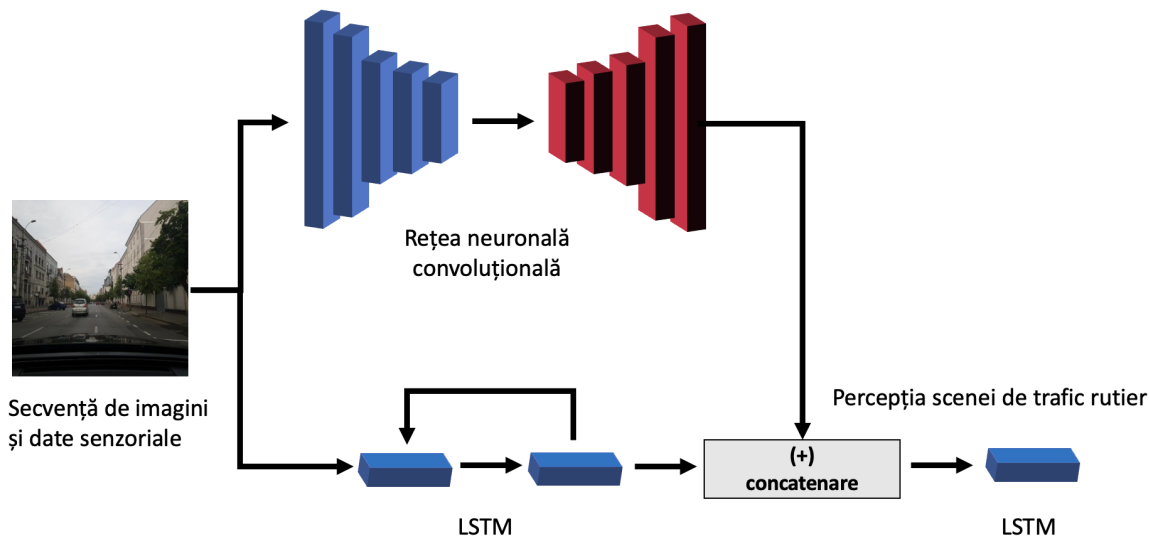


Figura 3. Structura initiala a rețelei propuse, bazata pe CNN si LSTM.

Reteaua prezentata in figura 3 are un total de 48.5 milioane de parametri. Rezultatul predictiei va oferi starea vehiculului pentru urmatoarea unitate de timp. Pentru antrenare se poate alege folosirea a unui numar de 10 observatii (10 imagini de intrare si 10 vectori care contin date senzoriale), iar predictia se va face pentru urmatoarele 5 unitati de timp / observatii (5 vectori care contin date senzoriale in urmatoarele 5 cadre). Aceste valori sunt configurabile si se pot schimba ulterior.

Pentru aceasta abordare, imi propun sa utilizez diferite configuratii pentru structura rețelei convoluționale pentru a obtine mai multe iesiri ale rețelei. Partea de antrenare a unei astfel de solutii se pot folosi si rețele pre-antrenate (prin initializarea ponderilor neuronilor cu valori din rețele pre-antrenate, in detrimentul initializarii cu valori aleatorii).

In etapa a doua a proiectului imi propun sa dezvolt aceasta abordare si sa o evaluez folosind metricele existente pentru a determina robustetea si acuratetea solutiei propuse. Totodata, rețeaua propusa as vrea sa o modific pentru a genera predictii pentru segmentarea semantica a scenei sau fluxul optic cu scopul de a determina informatii despre restul participantilor din trafic.

4. Implementarea sistemului de procesare pe sisteme desktop

Sistemul de procesare pentru sistem desktop este implementat folosind bibliotecile software open-source disponibile pentru antrenarea rețelelor de inteligența artificială și procesare de imagini. Astfel, pentru a implementa o rețea neuronală convoluțională pentru analiza scenei de trafic am folosit bibliotecile software TensorFlow [13], Keras [14], dar și PyTorch [15]. Aceste pachete software au fost realizate de Google (TensorFlow) și Facebook (PyTorch) și reprezintă momentan standardele folosite și în industria software pentru astfel de proiecte bazate pe inteligența artificială.

Am folosit TensorFlow pentru a realiza structura rețelei neuronale și apoi pentru a antrena rețeaua. Acelasi software este folosit și pentru a genera predicții folosind rețeaua antrenată anterior. Tot prin intermediul TensorFlow am implementat și partea de rețea neuronală recurentă (RNN), de tip LSTM ("Long Short Term Memory") care va genera predicții asupra stării curente a vehiculului și a scenei de trafic. Inițial, partea de segmentare semantică și de percepție a scenei de trafic am implementat-o folosind PyTorch, ulterior am realizat-o tot prin TensorFlow pentru a avea o singură rețea neuronală conform schemei prezentate în propunerea de proiect.

Totodată, pe sistemul desktop este instalat și software-ul Carla cu care se pot obține date din simulatorul 3D (imagini și măsurători senzoriale).

Publicatii

Razvan Itu, Radu Danescu, "Part-Based Obstacle Detection Using a Multiple Output Neural Network", Sensors, Vol. 22, No. 12, 2022, Art. No. 4312. [ISI – Q1, zona roșie]

Concluzii

În cadrul etapei inițiale am reușit crearea unor baze de date din cele existente și achiziția unor imagini noi folosind echipamentele disponibile în laboratorul de cercetare. Structura rețelei folosite pentru percepția scenei a fost folosită într-un sistem de segmentare semantică a scenei și a fost publicată o lucrare de jurnal. Structura aceleiasi rețele de percepție va fi folosită pentru modulul de codificare în noua rețea neuronală bazată pe "LSTM" (model recurent) care va oferi predicții asupra stării următoare a vehiculului.

Bibliografie

[1] - Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, Vladlen Koltun, CARLA: An Open Urban Driving Simulator, 2017, arXiv: 1711.03938

[2] - O. Ronneberger, P. Fischer, T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer, Vol. 9351, pp. 234-241, 2015.

[3] - K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition", Computer Vision and Pattern Recognition, pp. 770-778, 2016.

[4] - Razvan Itu, Radu Danescu, "Part-Based Obstacle Detection Using a Multiple Output Neural Network", Sensors, Vol. 22, No. 12, 2022, Art. No. 4312.

- [5] - Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., et al., 2016. End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316
- [6] - H. Xue, D. Q. Huynh, M. Reynolds, "SS-LSTM: A Hierarchical LSTM Model for Pedestrian Trajectory Prediction," 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 2018, pp. 1186-1194.
- [7] - Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Li, F.F., Savarese, S., 2016. Social Istm: Human trajectory prediction in crowded spaces, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 961–971.
- [8] - Z. Gu, Z. Li, X. Di, R. Shi, "An LSTM-Based Autonomous Driving Model Using a Waymo Open Dataset", Applied Sciences, 2020, Vol. 10, Art. No. 2046.
- [9] - Felipe Codevilla, Matthias Mueller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In 2018 IEEE International Conference on Robotics and Automation (ICRA), pages 1–9. IEEE, 2018.
- [10] - R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In International Conference on Machine Learning, pages 1310–1318, 2013.
- [11] - S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.
- [12] - H. Xue, D. Q. Huynh, M. Reynolds, "SS-LSTM: A Hierarchical LSTM Model for Pedestrian Trajectory Prediction," 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 2018, pp. 1186-1194.
- [13] - TensorFlow, accesat noiembrie 2022, online la adresa: <https://www.tensorflow.org/>
- [14] - Keras, accesat noiembrie 2022, online la adresa: <https://keras.io/>
- [15] - PyTorch, accesat noiembrie 2022, online la adresa: <https://pytorch.org/>

Director proiect,

Sl.Dr.Ing. Razvan Itu