

Detekcija voća

Laura Petan, Tea Teskera, Luko Lulić, Katarina Lukašević, Lorena Lazar

Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, Zagreb, Hrvatska

{laura.petan, tea.teskera, lukolulic, katarina.lukasevic, lorena.lazar}@fer.hr

Sažetak— Problem detekcije voća od značaja je za područja agronomije, edukacije i trgovine. Dosadašnji radovi koji su se bavili ovom temom postigli su izvrsne rezultate korištenjem neuronskih mreža baziranih na regijama (R-CNN). U ovom radu isprobano je više modela dubokog učenja te neke od klasičnih metoda strojnog učenja. Fokus rada je model YOLOv8 koji predstavlja napredak u domeni računalnog vida. Zbog dobrog performansa se koristi u većini zadataka iz tog područja uključujući detekciju objekata u stvarnom vremenu. Modeli R-CNN zahtijevaju više vremena, no i dalje imaju široku primjenu u spomenutom području. Značaj ovog rada je u usporedbi uspješnosti detekcije navedenih modela. Korišten je javno dostupni skup podataka koji okuplja 3 vrste voća, a jedan podatak je predstavljen slikom objekata i pozadine te .xml datotekom s koordinatama i klasama objekata. Rezultati detekcije modela YOLO su uspoređeni s rezultatima Faster R-CNN-a, KNN-a te SVM-a. Očekivano, YOLO postiže značajno najbolje ocijene, zatim klasične metode, a konvolucijske mreže su rangirane najslabije.

Ključne riječi – detekcija voća, lokalizacija voća, klasifikacija voća, YOLO, faster R-CNN, SVM, konvolucijske neuronske mreže, k-NN

I. UVOD

Značajan razvoj računarске znanosti i tehnologija računalnog vida omogućio je rješavanje mnogobrojnih problema. Računalni vid jedno je od područja umjetne inteligencije koje omogućuje računalu razumijevanje vizualnog svijeta. Računala mogu koristiti digitalne slike i modele dubokog učenja za točnu identifikaciju i klasifikaciju objekata te sukladno tomu obraditi informacije za dobivanje željenog rezultata.

U agrikulturi, obrada slike koristi se sve više u svrhu automatiziranja procesa. Automatski sustavi koriste se za pregled progresa usjeva, bolesti usjeva te za prepoznavanje voća i povrća. Prepoznavanje te klasifikacija voća može se koristiti u mnogim primjenama u svakodnevnom životu. Primjerice, u sustavima naplate u supermarketima gdje se mogu koristiti umjesto manualnih bar kodova, također i kao alat u obrazovanju, specifično za malu djecu te pacijente s Down sindromom.

U ovome radu provedena je detekcija na slikama voća. Detekcija obuhvaća lokalizaciju mjesta objekata i klasifikaciju tih objekata. Za detekciju voća korišteni su algoritmi YOLOv5 i Faster R-CNN što spada u moderne metode dubokog učenja. Oba algoritma koriste konvolucijske neuronske mreže. Dodatno su korištena dva tradicionalna pristupa strojnog učenja: stroj potpornih vektora (engl. *support vector machine*)

i k–najbližih susjeda (engl. *k-nearest neighbors*) za klasifikaciju voća. Na kraju su napravljene usporedbe i prikazani rezultati svega navedenog.

U nastavku rada prikazan je pregled postojećih pristupa za uspješnu detekciju voća. U trećem poglavlju opisana je metodologija rada, a u četvrtom poglavlju prikazani su dobiveni rezultati. Peto poglavlje sadrži diskusiju dobivenih rezultata, a zaključak rada je naveden u šestom poglavlju.

II. PREGLED POSTOJEĆIH PRISTUPA DETEKCIJE VOĆA

Detekcija objekata je osnovni zadatak u području računalnog vida. Obuhvaća pronalazak mjesta gdje se nalazi određeni objekt te klasificiranje pronađenog objekta u jednu od klasa. Zbog navedenog se detekcija voća može razdvojiti u dva koraka:

- lokalizacija voća
- klasifikacija voća

Većina radova koji se u današnje vrijeme fokusiraju na klasifikaciju voća koriste u širokom rasponu neuronske mreže. Na primjer, u radu [2] dizajniran je model za prepoznavanje povrća temeljen na obradi slike i raznim postupcima korištenja računalnog vida. Autori uspoređuju 24 različitih vrsta povrća koristeći 3924 slike iz baze podataka. Prvotno je obavljeno pretprocesiranje slika normalizacijom te je promijenjena veličina slika u bazi podataka. Potom je implementiran model učenja konvolucijske neuronske mreže (CNN) koji je prepoznavao povrće korištenjem karakteristika boje i teksture te postigao približno 95,50 % točnosti. U radu [3] izdanom 2019. godine autori su se usredotočili na automatsku klasifikaciju maslina. Za potrebe rada fotografirano je 2800 plodova maslina sedam različitih sorti. Korišteno je šest različitih arhitektura konvolucijskih neuronskih mreža, no na kraju je Inception-ResNetV2 dao najbolje rezultate. Ovaj model postiže klasifikacijsku točnost od 95,91%. U radu [4] izdanom 2020. godine autori su radili sa specifičnom bazom podataka koja je sadržavala četiri različite vrste voća te su obrađivali slike iz zadane baze različitim algoritmima za obradu slike. U ovom radu korišteni su klasifikatori BPNN, SVM i CNN, pri čemu CNN pruža najveću točnost od 90% u usporedbi s drugim klasifikatorima. U području detekcije i prepoznavanja objekta postignuti su izvrsni rezultati korištenjem konvolucijskih neuronskih mreža temeljenim na regijama (engl. *Region Based Convolutional Neural Networks, R-CNN*) [5]. Projekti koji su koristili navedeni algoritam, postigli su rezultate od 60% do

70% točnosti [6]. Algoritmi iz porodice YOLO koriste se u području računalnog vida za detekciju objekata na slici. Vrlo su rašireni u uporabi s obzirom na to da svojom brzinom omogućuju detekciju objekata u stvarnom vremenu. Zbog toga se primjenjuju za rješavanje problema detekcije objekata u autonomnoj vožnji, analizi sporta, praćenja jedinki iz određene životinjske populacije i mnogih drugih [8].

III. METODOLOGIJA

Za ovaj je rad odabran javno dostupan skup podataka sa stranice *Kaggle* [1]. Podatkovni skup korišten za izradu ovog rada sastoji se od ukupno 300 slika na kojima se nalazi voće i 300 .xlm datoteka u kojima su zapisane koordinate piksela na slici gdje se nalazi pojedino voće i kategorija (klasa) kojoj to voće pripada. U ovom radu voće je moguće klasificirati u jednu od 3 klase: banana, jabuka i naranča. Podatkovni skup je podijeljen na skup od 240 slika koje se koriste kod učenja i 60 preostalih slika koje se koriste za testiranje uspješnosti detekcije i klasifikacije modela. Na slici 1. prikazano je nekoliko slika iz korištenog skupa. Na nekim slikama nalazi se raznovrsno voće (prva slika), na nekim slikama nalazi se samo po jedno voće (druga slika), na nekim slikama nalazi se više istog voća (treća slika), a na nekim slikama se dodatno nalazi voće koje ne pripada u nijednu od navedene 3 klase (četvrta slika). Slike su dosta jednostavne te se očekuju dobri rezultati.



Slika 1 Prikaz različitih slika voća

Kako bi se postigla što bolja točnost modela, slike su predobrađene po uzoru na navedene radove. Odabrano je Postupak detekcije proveden je uz pomoć algoritama YOLOv5 i Faster R-CNN. Detektirane slike objekata je prije klasifikacije potrebno obraditi tako da sve budu istih dimenzija. Odabrane su dimenzije 60x60 što je dovoljno za ovaj jednostavan podatkovni skup. Za klasifikaciju slika pomoću tradicionalnih pristupa korišteni su algoritmi: stroj potpornih vektora i k – najbližih susjeda.

A. YOLO

Akronim YOLO stoji za *You Only Look Once*, što je referenca na činjenicu da slika treba proći kroz mrežu algoritma iz porodice YOLO samo jednom kako bi objekti na njoj bili detektirani i klasificirani. Detekcija objekata u YOLO algoritmima implementirana je kao regresijski problem te se u njima koristi konvolucijska neuralna mreža (CNN) [8].

Osnovni princip rada YOLO algoritama svodi se na tri metode: metoda rezidualnih blokova, regresije graničnog okvira te presjeka nad unijom. Metodom rezidualnih blokova slika je podijeljena u nekoliko rešetki. Svaka rešetka je istih dimenzija, a svaka ćelija rešetke će detektirati granične okvire koji se pojavljuju unutar te ćelije. Zatim se regresijom graničnog okvira dobivaju vjerojatnosti da je objekt unutar pojedinog graničnog okvira. Na kraju se metodom presjeka nad unijom dobiva

numerička mjera slaganja graničnog okvira dobivenog predikcijom s temeljnom istinom (engl. *ground truth*) [9].

Prva verzija YOLO algoritma objavljena je 2015. godine. Njegove glavne slabosti bile su nemogućnost detekcije novih ili neobičnih oblika te poteškoće pri detekciji malih objekata u slici, poput grupe ljudi na stadionima [10].

Usljedile su mnoge iteracije iz porodice YOLO, a u nastavku će se pobliže razmotriti YOLOv8.

YOLOv8 najnoviji je algoritam iz porodice YOLO. Pušten je u javnost 10. siječnja 2023. godine, a razvila ga je tvrtka Ultralytics. Model YOLOv8 dolazi u 5 veličina, od kojih je najmanji YOLOv8n s 3.2 milijuna parametara, a najveći je YOLOv8x sa 68.2 milijuna parametara [12]. Bolji je od prethodnih YOLO algoritama po metrikama brzine, točnosti i efikasnosti. Kao bazu arhitekture YOLOv8 koristi modificirani CSPDarknet53. Autori YOLOv8 modela su ga trenirali na MS COCO skupu podataka [11]. U YOLOv8 algoritmu nalaze se tri velike promjene u usporebi s prethodnim iteracijama, a to su neusidrena detekcija (engl. *anchor free detection*), introdukcija C3 konvolucija te mozaička augmentacija podataka. Tijekom treniranja, YOLOv8 koristi mozaičku augmentaciju podataka što je relativno jednostavna operacija nad slikovnim podacima kod koje su četiri različite slike spojene zajedno i korištene kao ulaz u model. Ova operacija omogućava modelu da zapravo uči objekte, a ne samo pozicije specifičnih objekata [12].

B. R-CNN

Generalno, konvolucijske mreže postižu dobre rezultate pri radu s višedimenzionalnim podacima, tj. slikama. Međutim, problem detekcije objekata je specifičan jer je potrebno izvršiti dodatan korak prije klasifikacije. Budući da nam nije poznato gdje na slici se nalazi objekt od interesa, sliku je prvo nužno podijeliti na više regija. Objekt također može biti proizvoljno velik ili malen pa regije moraju biti dovoljno sitne za preciznu detekciju. Kada bi ovaj postupak – dijeljenje u regije i određivanje relevantnosti svake – radili običnim CNN modelom, on bi trajao jako dugo te zahtijevao previše računalnih resursa. Algoritam R-CNN rješava taj problem uvođenjem konstantnog broja regija. Slika se prvo podijeli u velik broj regija, zatim se regije pohlepno algoritmom rekurzivno grupiraju u veće dijelove dok se ne pronađe konačnih 2000 koje će se koristiti kao ulaz u neuronsku mrežu. CNN na izlazu daje 4096-dimenzionalni vektor gdje svaka komponenta predstavlja jednu značajku. Taj vektor se zatim predaje SVM-u. SVM daje odluku o prisutnosti značajke u regiji te također daje pomak u odnosu na granice regije. Ovaj postupak ubrzava detekciju objekta, no njegovo trajanje i dalje nije zanemarivo – za obradu jedne slike potrebno je oko 50 s.

Kao nadogradnju na prethodni algoritam, predložen je brzi R-CNN (engl. *Fast R-CNN*) [7]. Ovaj algoritam ne vrši podjelu na regije, već konvolucijskoj mreži predaje cjelovitu sliku. Mreža zatim sama radi mapu značajki na temelju koje se algoritmom selektivnog pretraživanja određuju regije. Posljednji sloj mreže preuzima zadatak SVM-a. Ovaj pristup je značajno brži – obrada jedne slike traje manje od 1 s. Međutim, usporava ga biranje regija koje može produljiti vrijeme detekcije na 3 s.

Najnovije poboljšanje donosi algoritam brži R-CNN (engl. *Faster R-CNN*). On eliminira potrebu za selektivnim traženjem regija te umjesto toga taj zadatak prepušta neovisnoj mreži. Ostatak detekcije isti je kao kod R-CNN-a. Vrijeme potrebno za obradu jedne slike sada je smanjeno na oko 0.2 s.

C. Stroj potpornih vektora

Stroj potpornih vektora ili kraće SVM je popularan algoritam strojnog učenja razvijen 1990-ih koji daje veoma uspješne rezultate za probleme klasifikacije. SVM je generalizacija jednostavnijeg linearnog klasifikatora koji se naziva klasifikator maksimalne margine. Nelinearnost kod algoritma SVM postižemo korištenjem jezgri [13][13][7]. Kod izgradnje modela korištene su definirane vrijednosti jer su se one pokazale najboljima. Definirana jezgra je radijalna što razdvajanje nelinearnih podataka, a hiperparametar C postavljen je na 1.

D. K-najbližih susjeda

K-najbližih susjeda ili k-NN je jedan od jednostavnijih algoritama klasifikacije. On traži k susjeda koja su u skupu za treniranje najbliža trenutnom uzorku u skupu za testiranje te na temelju većinskog broja primjera iz neke klase dodjeljuje klasu promatranom uzorku. Uglavnom se najbliži susjedi traže izračunavanjem euklidske udaljenosti od testnog uzorka. Hiperparametar k, to jest broj susjeda, je postavljen na broj 4 jer se to pokazalo kao najbolje rješenje te je postignuta najveća točnost na ovom primjeru.

Za implementaciju korištenih algoritama i predobradu slike korištene su biblioteke scikit-learn, OpenCV, PyTorch i NumPy. Za izgrađene modele su ispisane neke mjere validacije kao što su odziv (engl. *recall*), preciznost (engl. *precision*) i F1-vrijednost (engl. *F1-score*) koja je kombinacije prethodne dvije. Također je prikazana točnost izgrađenih modela.

IV. REZULTATI

Mjere validacije koje su korištene za

U nastavku su za svaki izgrađen model detaljno prikazani i opisani dobiveni rezultati.

A. YOLOv8

Za YOLOv8 isprobano je tri od pet dostupnih veličina modela. Testiranjem se pokazalo da se smanjenjem dimenzija slika zadržava kvaliteta rezultata, a ubrzava vrijeme treniranja. Stoga su za treniranje korištene slike dimenzija 400x400 piksela. Kao početna točka korišten je predtreniran model jer možemo očekivati da objekti koje želimo detektirati imaju dovoljno sličnosti s objektima u općem skupu koji je korišten za predtrentiranje. Svi modeli trenirani su pomoću Google Colab koristeći Tesla T4 GPU.

Model veličine „S“ (small) treniran je u 75 epoha. Treniranje je ukupno trajalo 13 minuta i 48 sekundi. Veličina modela je 22.5 MB. Vrijeme inferencije je 2.7 ms.

Model veličine „L“ (large) treniran je u 100 epoha. Treniranje je ukupno trajalo 13 minuta i 25 sekundi. Veličina modela je 87.6 MB. Vrijeme inferencije je 8.4 ms.

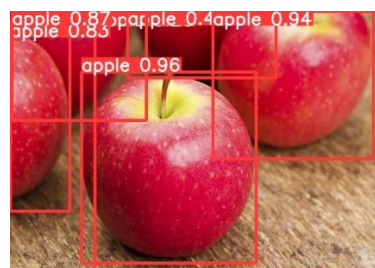
Model veličine „X“ (extra large) treniran je u 50 epoha. Treniranje je ukupno trajalo 17 minuta. Veličina modela je 136.7 MB. Vrijeme inferencije je 12.5 ms.

Rezultati za sva 3 navedena modela su prikazani u tablici 1. Sveukupno najbolje rezultate postiže najveći model. Konačan odabir modela osim o samoj kvaliteti detekcije i klasifikacije ovisi i o drugim praktičnim parametrima poput vremena treniranja, vremena zaključivanja i samoj veličini modela. S obzirom na to da točnost isprobanih modela ujednačeno raste s njihovom „cijenom“ konačan odabir ovisi o konkretnoj primjeni i mogućnostima.

Tablica 1. Prikaz mjera validacije za YOLOv8

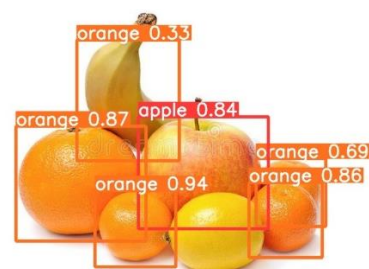
	Precision	Recall	mAP50	mAP50-95
„S“ (small)	0.837	0.855	0.926	0.728
„L“ (large)	0.942	0.784	0.93	0.749
„X“ (extra large)	0.874	0.934	0.962	0.77

Primjer uspješne detekcije i klasifikacije voća prikazan je na slici 2. Vidljivo je da su sve jabuke uočene te su točno klasificirane sa sigurnošću većom od 80%.



Slika 2. Uspješna detekcija i klasifikacija

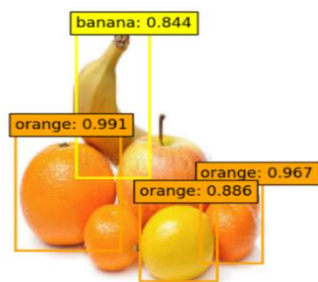
Primjer neuspješne detekcije i klasifikacije voća prikazan je na slici 3. Vidljivo je da je banana klasificirana kao naranča.



Slika 3. neuspješna detekcija i klasifikacija banane

B. Faster R-CNN

Model Faster R-CNN treniran je u 30 epoha te je treniranje trajalo 12,5 minuta. Korištene dimenzije slika su 224x224. Dobivena mjera mAP iznosi otprilike 56% na skupu za testiranje. Na slici 4. je vidljivo da je ovaj model uspješno klasificirao bananu iz prethodne slike, no nije uspio detektirati jabuku.



Slika 4. Primjer detekcije i klasifikacije modela Faster R-CNN

C. Stroj potpornih vektora

U tablici 2. vidljivo je da je F1-vrijednost najveća za klasu banana što je očito jer su jabuka i naranča slične po obliku, to jest oboje su okrugle, a banana je duguljasta. Najveću preciznost također ima klasa banana. To znači da od svih slika koje su klasificirane kao banana, njih 95% to uistinu i jesu. Odziv je najveći za klase banana i naranča te iznosi 88%. To znači da od svih slika na kojima se stvarno nalazi naranča, 88% slika je točno klasificirano. Najgora F1-vrijednost je dobivena za klasu jabuka. Točnost na skupu za treniranje iznosi 94,2%, a točnost na testnom skupu iznosi 86,3%.

Tablica 2. Prikaz mjera validacije za model izgrađen algoritmom SVM

KLASA	preciznost	odziv	F1-vrijednost	količina
jabuka	0.83	0.83	0.83	35
banana	0.95	0.88	0.91	40
naranča	0.82	0.88	0.85	42

D. K-najbližih susjeda

U tablici 3. je vidljivo da je F1-vrijednost ponovo najveća za klasu banana što je vjerojatno zbog njezinog oblika. Najmanju F1-vrijednost ima klasa jabuka. Preciznost je najveća za klasu banana, a najmanja za klasu naranča. Odziv je najveći za klasu banana, a najmanji za klasu jabuka. Ukupna na točnost na skupu za treniranje iznosi 94,2%, a točnost predviđanja na testnom skupu iznosi 82,9%.

Tablica 3. Prikaz mjera validacije za model izgrađen algoritmom k-NN

KLASA	preciznost	odziv	F1-vrijednost	količina
jabuka	0.81	0.74	0.78	35
banana	0.90	0.88	0.89	40
naranča	0.78	0.86	0.82	42

V. ZAKLJUČAK

Najbolji rezultati postignuti su za model „X“ YOLOv8. Ocjena mAP50 iznosi 0,962, a mAP50-95 iznosi 0,77. Ocjene R-CNN modela su niže od očekivanih, mAP poprima vrijednost između 50% i 60%. Ove rezultate možemo objasniti jednostavnošću ulaznog skupa. Konvolucijske mreže trebaju biti trenirane na velikom skupu podataka kako bi dale dobre rezultate. Samo 260 ulaznih primjera nije bilo dovoljno da bi uspješno naučile klasificirati. Klasični modeli, KNN i SVM postižu F1- vrijednost veću od 80% što nadmašuje performanse

R-CNN-a. Najbolji rezultati općenito su postignuti za klasu banana. Trivijalno je vidjeti da je uzrok tomu to što je ta klasa najmanje slična ostalima. Budući da spomenuto voće ima vrlo specifičan oblik, mrežama je lakše naučiti objekte koji pripadaju toj klasi.

Osim uspješnosti detekcije i klasifikacije treba promatrati i vrijeme treniranja te donošenja odluke. Model YOLOv8 je značajno brži od konvolucijskih modela pri donošenju odluke. Treniranje najvećeg modela je trajalo 17 min, dok je odluka za jednu sliku donesena u nešto više od 12 ms. Fast R-CNN model treniran je u 12,5 min, ali mu treba više od sekunde da obradi sliku. Klasični modeli su provodili samo klasifikaciju, bez detekcije pa njihova vremena nisu usporediva.

Iz dobivenih rezultata može se zaključiti da model YOLO ima najbolje performanse promatrajući dvije najvažnije metrike (ocjenu mAP te vrijeme donošenja odluke). Ovi rezultati mogu biti od značaja u mnogim područjima koja koriste računalni vid. U nastavku istraživanja valjalo bi usporediti različite verzije YOLO modela te izvršiti istraživanje na opširnijem skupu podataka.

POPIS LITERATURE

- [1] M. Buyukkinaci, "Fruit Images for Object Detection". 2018. [Online] <https://www.kaggle.com/datasets/mbkinaci/fruit-images-for-object-detection> (Posjećeno: 9.3.2023.)
- [2] P. S. Dutch and K. Jayasimha, "Intra class vegetable recognition system using deep learning," in Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 602–606, Maisamaguda, India, June 2020.
- [3] J. M. Ponce, A. Aquino, and J. M. Andújar, "Olive-fruit variety classification by means of image processing and convolutional neural networks," IEEE Access, vol. 7
- [4] R. Yamparala, R. Challa, V. Kantharao, and P. S. R. Krishna, "Computerized classification of fruits using convolution neural network," in Proceedings of the International Conference on Smart Structures and Systems (ICSSS), Chennai, India, July 2020.
- [5] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, and A.W.M. Smeulders: *Selective Search for Object Recognition*
- [6] Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik UC Berkeley: Rich feature hierarchies for accurate object detection and semantic segmentation Tech report
- [7] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun: *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*
- [8] J. Terven, D. Cordova-Esparza, A *Comprehensive Review of YOLO: From YOLOv1 and Beyond*, arXiv, 2023.
- [9] G. Karimi, *Introduction to YOLO Algorithm for Object Detection*, section.io, <https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/> (Posjećeno: 20.5. 2023.)
- [10] Z. Keita, *YOLO Object Detection Explained*, datacamp.com, <https://www.datacamp.com/blog/yolo-object-detection-explained> (Posjećeno: 20.5. 2023.)
- [11] M. Moin, *Unleashing the Power of YOLOv8: A Comprehensive Guide to the Real-Time Object Detection Model*, medium.com <https://medium.com/augmented-startups/unleashing-the-power-of-yolov8-a-comprehensive-guide-to-the-real-time-object-detection-model-af71014a5e7a> (Posjećeno 20.5. 2023.)
- [12] M. Krishnakumar, A Gentle Introduction to YOLOv8, wandb.ai <https://wandb.ai/mukilan/wildlife-yolov8/reports/A-Gentle-Introduction-to-YOLOv8--Vmlldzo0MDU5NDA2> (Posjećeno: 20.5. 2023.)
- [13] James, G., Witten, D., Hastie, T., Tibshirani, R., *An Introduction to Statistical Learning: with Applications in R*, 2. izdanje, Springer, 2013

