Duck hunt igra kontrolisana mobilnim uređajem

Nenad Mišić
Fakultet Tehničkih Nauka, Novi Sad

Uvod

Korišćenje tehnika obrade digitalne slike pri implementaciji kopije poznate "Duck Hunt" igre razvijene od strane Nintendo korporacije za njihovu NES platformu.

Igra se pokreće na računaru dok se kontroliše pomoću mobilnog uređaja povezanog na lokalnu mrežu sa tim računarom.

Ideja je napraviti program koji procesira sliku koju dobija od mobilnog telefona, zaključuje da li je telefon uperen prema nekoj od ptica i, ukoliko jeste, ubija pticu.

Cilj

Kako se kupovinom originalne igre za Nintendo, dobijao i kontroler kojim se upravljalo igrom, cilj ovog projekta je napraviti što sličnije iskustvo u igranju igre uz pomoć tehnika obrade digitalne slike naučenih na kursu "Soft computing".

Jedan od problema na koji se nailazi je to što interakcija mobilnog uređaja i računara na kom je pokrenuta igrica mora biti u realnom vremenu sa što manjom latencijom u komunikaciji da bi korišćenje bilo glatko. Tehnike primenjene u implementaciji ovog problema moraju performansama da se prilagode ovom zahtevu.

Skup podataka

S obzirom da je problem koji se rešava vrlo usko specifičan, skup podataka je morao biti ručno prikupljen i anotiran.

Slike na kojima je program treniran su telefonom uslikani trenuci u igri, sa kojih je prvo izdvajan ekran.
Na slikama ekrana, ručno su označavane ptice kao regije od interesa.
Nakon toga, slika ekrana se deli na 15 kolona I 10 redova i čuvaju se blokovi veličine 2x2 ćelije.
Svaki 2x2 blok koji preseca regiju od interesa označen je kao potencijalna ptica, dok sve ostale, koje ne presecaju regiju od interesa, bivaju označene kao sigurni delovi pozadine.
Poslednji korak je ručno pregledanje potencijalno pozitivnih slika I izbacivanje onih koje nisu odgovarajuće (recimo slike na kojima se vidi samo par piksela ptice).

Korišćeno je 80 početnih slika iz kojih je dobijeno oko 2000 pozitivnih i 8000 negativnih slika. Nad njima se dalje vrši augmentacija i izdvaja "sample" negativnih slika kako bi obe klase imale isti broj članova.

Metode

Sa ulazne slike se uz oslanjanje na Canny Edge Detector detektuju ivice na slici, i iz njih se izvlači kontura koja veličinom odgovara konturi ekrana. Takođe, na originalnoj slici se pamte koordinate centra, jer to predstavlja mesto na kom se u trenutku pucanja nalazi nišan.

Zatim se pronalaze koordinate centra na isečenoj slici ekrana i oko centralne tačke se seče kvadrat veličine 100px * 100px.

Isečena slika se dalje prosleđuje klasifikatoru koji ume da prepozna pticu. U ovom projektu korišćeno je više tipova klasifikatora kako bi se pronašlo koji daje najbolje rezultate za dati problem. Korišćeni klasifikatori su sledeći:
1. KNN sa oslanjanjem na HOG Feature Extractor
2. SVM sa oslanjanjem na HOG Feature Extractor
3. CNN

Ukoliko je izlaz iz klasifikatora pozitivan (detektovana je ptica), pomoću PyAutoGUI biblioteke izvrši se klik kojim se ubija ptica na zadatoj poziciji.

Rezultati

Skup podataka je podeljen na trening i test skup u razmeri 80:20.
Poređenjem rezultata primenom sva tri implementirana klasifikatora nad testnim skupom podataka izvučen je zaključak da se CNN (Konvoluciona neuronska mreža) bolje pokazala od ostala dva klasifikatora. Rezultati poređenja preciznosti prikazani su u tabeli ispod.

	Trening	Test
KNN	0,8235	0,7727
SVM	0,9992	0,9354
CNN	0,9284	0.9809

Zaključak

Iz dobijenih rezultata možemo zaključiti da se KNN klasifikator dosta lošije pokazao od SVM klasifikatora uprkos tome što oba koriste identičan pristup ekstrakciji osobina sa slike (HOG Feature Extractor). To je bilo donekle i očekivano ponašanje. Sa druge strane, vidimo da SVM nije u stanju da dovoljno dobro vrši klasifikaciju nad ovakvim podacima (ako pretpostavimo da tačnost od 93% nije dovoljna). Razlog za ovo je što se pozitivne i negativne slike jako malo razlikuju (lako je pomešati pticu, drvo i žbun, recimo), pa je potreban nešto "pametniji" pristup, poput CNN.