En ML-plattform som kan

Av Ali Osman Örs, NXP Semiconductors och David Steele, Arcturus

Ali Osman Örs är AI-strateg på NXP Semiconductors med 20 års erfarenhet inom halvledarindustrin efter en ingenjörsutbildning i Ottawa, Kanada. Specialiteterna är ML och datorseende. Tidvis har han ansvarat för NXP:s strategier inom AI för förarassistans och autonoma fordon. Idag ansvarar han för AI-edge. Han rekryterades från CogniVue som utvecklar systemkretsar för AI-bildanalys.

David Steele är innovationschef på Arcturus inom edge-AI och datorseende. David har under sina 20 år i branschen bland annat ansvarat för att introducera nya teknikområden som Linux, IoT, edge och röst- och videokommunikation. David har utbildning från Ryerson University i Toronto.

AI-modeller kan vara imponerande i experimentsituationer, men för praktiska tillämpningar krävs ytterligare några processteg som inte syns i labbet: indata och utdata från modellen kan inte användas som de är. En komplett lösning som täcker hela processkedjan kan ha stor nytta av en flexibel plattform för hårdvara och firmware utvecklad av NXP och Arcturus.

Maskininlärning baserad på DNN (djupa neuronnät) har gett många imponerande experimentella resultat, i synnerhet inom bildtolkning för ansikts- och objektigenkänning. Den tidiga utveckling inom DNN, liksom de första skarpa tillämpningarna, har skett på kraftfulla molnservrar. Det var nödvändigt för att kunna leverera den tunga beräkningskapacitet som krävs.

Databehandling på molnserver är dock ofta inte särskilt praktisk. Lokala datorresurser har mycket lättare att uppfylla krav på bandbredd och latens.

Här följer ett hypotetiskt, men högaktuellt exempel. Banker kan ha kameraövervakning vid sina bankomater. Och under pandemin har det funnits påbud om att hålla avstånd. En del banker har därmed fått den goda idén att använda sina kameror till att automatiskt kontrollera att köande och folk runtomkring står tillräckligt långt ifrån varandra. Banken kanske till och med vill se till att endast kunder som bär mask får använda automaten eller komma in i lokalen där bankomaterna står.

Ett sätt att AI-processa videoströmmen från kameran vore att tanka upp den direkt till molnet. Men det kan potentiellt vara en väldigt dyr lösning. Kanske till och med praktiskt omöjligt någon annanstans än i en tät stadskärna.

Det innebär dessutom en fördröjning, vilket betyder att det blir svårt för systemet att svara på rimlig tid när det anländer nya kunder eller att hinna kontrollera avstånden när besökarna flyttar sig runt i rummet. Om videodata istället kunde bearbetas lokalt skulle det ge möjlighet för systemet att reagera snabbt. Det kräver förstås en hårdvara kraftfull nog att kunna implementera en komplett processkedja för bildigenkänning från rådata till analys.

Modellval

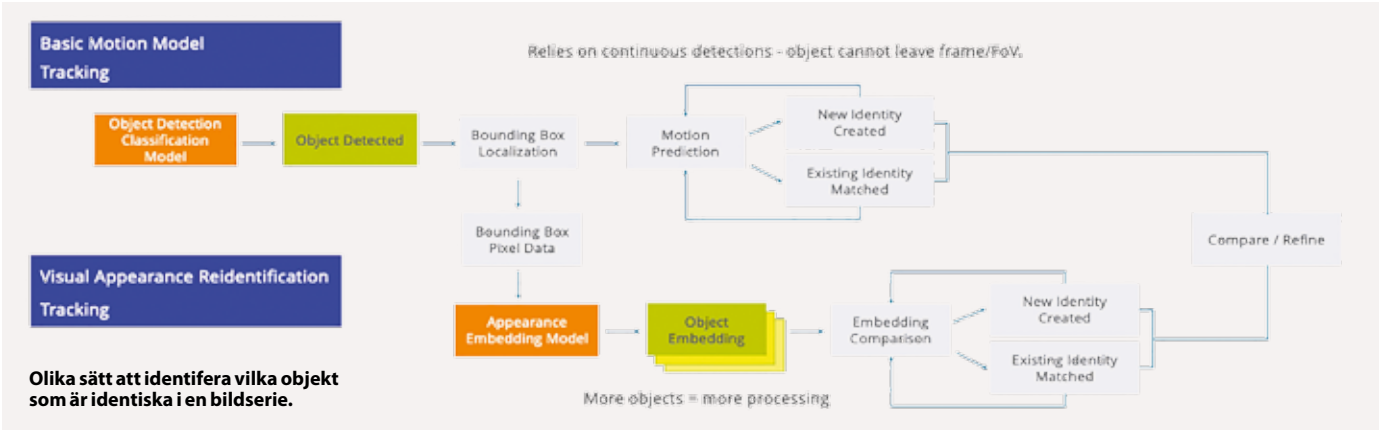
Vad gäller högpresterande edge-hårdvara för DNN finns numera ett brett utbud. En möjlighet är att välja en grafikprocessorer (GPU:er) med en arkitektur delvis modifierad för högre DNN-genomströmning. Men inget klår en dedikerad NPU (neural processing unit). Den ger överlägset mest DNN-prestanda per watt.

Viktigt är att välja hårdvara med stor inbyggd flexibilitet, och att inte bara gå efter deras prestanda för de typiska standardiserade DNN-prestandamåtten, som ImageNet och MobileNet. Det är dessutom typiskt nödvändigt att förbearbeta bilddatan till en form som passar tillämpningen bättre. Och så måste de djupa neuronnäten dessutom typiskt finjusteras för att hantera specifika krav.

Exemplet med bankomatövervakning kan användas för att illustrera nämnda be-

Det finns många sätt att täcka ansiktet på. De måste analyseras separat.

Om bilden tas ur grod- eller fågelperspektiv påverkas säkerheten i klassificeringen.



Olika sätt att identifera vilka objekt som är identiska i en bildserie.

anpassas till verkligheten

hov av finjusteringar och förbearbetning. Maskininlärningsexperten Arcturus Networks har analyserat testvideoinspelningar från en bankklient.

Små, slutna bankomatutrymmen betyder typiskt att kameravinklarna blir tvära vilket i inspelningarna ledde till fler felklassificeringar, särskilt när människor befann sig i rörelse framför kameran. Andelen korrekta klassificeringar kunde sjunka från drygt 98 procent när nästa kunds ansikte tydligt syntes i bild, till mindre än 40 procent när kameran bara kunde se toppen av huvudet och en liten del av ansiktet.

Munskydd ökar förstås komplexiteten ytterligare. Det räcker förresten inte att träna nätet att identifiera ”människa med mask”. Också hjälm och andra ansiktsskydd är en form av personlig skyddsutrustning (PPE) som förekommer. Även andra typer av ansikts- och huvudtäckning behöver kunnas sortera i sina egna klasser.

Det går att tänka sig att ställa ytterligare krav på systemet, exempelvis att det ska kunna ha förmågan att upptäcka avvikande beteenden. Som människor som hänger kvar i lokalen utan att ha något ärende.

Ett annat intressant fall är när motivet inte hela tiden är synligt. Personer kan vandra in och ut ur kamerans synfält. Det skapar en ny typ av krav: förmågan att spåra ett visst motiv över tid. Det räcker inte med att bara kunna identifiera kategorier som ”maskbärare” och ”personer som håller avstånd”. De nya kraven behöver egna anpassade algoritmer för att förbearbeta data. De kräver även en anpassning av hur modellen används.

Det är svårare att bedöma om en person bär PPE i livevideo än i labbexperiment, eftersom personer kan vara skymda, eller byta kroppsställning. Därför kommer korrektheten i klassificeringen att variera. Ett sätt att förbättra noggrannheten är att använda flera bilder för en och samma klassificering. Men för det krävs att individer kan spåras mellan bilderna.

Rörelsespårning i sig är i princip inte extremt beräkningskrävande. Problemet är att följa personer mellan bilder. Det försvåras av att personer försvinner ur bild eller blir tillfälligt skymda. De uppfattas lätt som nya personer som dyker upp istället för att kopplas ihop med redan sedda personer.

Ett sätt att hantera spårningen av individer är att addera extra data till representationen av identifierade bildobjekt. Det kallas inbäddningar (embeddings). Sådana används exempelvis ofta i databehandling av språk. Fraser och ord märks upp med vektorer som används för att konstatera att fraser och ord har liknande betydelser.

För bankomatövervakningen använder vi en inbäddning som inte bara anger vilken kategori objektet sorterades in i, utan även hur det visuellt såg ut och var i bilden som det påträffades. Positionen anges av en fyrhörning som ramar in objektet (en bounding box). Bilden av objektet i boxen analyseras på bildpunktsnivå för att identifiera olika karaktärsdrag (features) i bilden. De anges numeriskt i en vektor som alltså utgör en representation av objektets synliga former, färger och strukturer.

En fördel med inbäddningar är att de kan delas mellan flera olika kameror för att öka noggrannheten eller för att kunna täcka en större yta.

Inbäddningarna kan även exempelvis användas som sökindex i ett arkiv över sedda objekt, kanske som referenser i en bevakningslista för analys som kan göras senare offline.

Den databehandling som görs för att kunna följa individer mellan bildrutor innebär ett overhead vilket påverkar genomströmningen. I bästa fall ryms inte fler personer i rummet än att de alla kan följas. I andra fall kanske det behöver adderas fler SoC:er för databehandlingen.

Flexibel arkitektur

Något som vore högst önskvärt vore alltså hårdvara med flexibiliteten att kunna hantera samtliga de olika moment som maskininlärning för datorseende kan stöta på i ett fall ur verkligheten.

För detta har Arcturus valt NXP:s systemkrets i.MX8M Plus. Den består av ett antal beräkningskärnor som Arcturus flexibelt kan sätta ihop till en pipeline för datorseende. Den är enkel att anpassa. Varje steg i databehandlingen representeras av en nod.

Exempelvis kan en nod ansvara för inferenser eller för att hantera förprocessning eller efterprocessning. En annan nod kan hantera användningen av en extern eller distribuerad tjänst.

Systemlösningen liknar lite grand det ▶▶

containeriserade tillvägagångssätt som används i cloud computing. Men här anpassad till de resursbegränsningar som gäller för edge computing.

Varje nod fungerar som en mikrotjänst. Noderna kopplas samman via synkroniserade seriella dataströmmar och bildar tillsammans en komplett pipeline för datorseende. Den kan hantera allt från bildinsamling till lokal styrning av periferiutrustning på plats.

Enkla tillämpningar kanske klarar sig med noder som körs i en pipeline på en och samma fysiska resurs. En mer komplex pipeline kan behöva distribuera sina noder över flera CPU:er, GPU:er och NPU:er eller kanske över en eller flera i.MX 8M Plus-processorer. Eller kanske till och med över molnet.

Arkitekturen tillåter att en pipeline orkestreras under körning, alltså att den kan ombildas efter förändrade behov i tillämpningen.

Detta hjälper till att säkra investeringen för framtida edge-tillämpningar. Varje nod är containeriserad och därmed är det enkelt att byta ut en del av systemet i taget. Exempelvis kan en inferensmodell uppdateras utan att resten av systemet rubbas, även om det krävs ändrade attribut i modellen.

Nämnda pipeline-arkitektur kan byggas i utvecklingsmiljön Arcturus Brinq Edge Creator SDK. Detta inkluderar möjligheten att skala upp AI-prestanda genom att använda mer än en enda fysisk processor.

En i.MX 8M Plus kan generera inbäddningar för djupa neuronnät (DNN) på en eller flera i.MX 8M-enheter som kan ta indata från en eller flera kameror. Enheterna kan enkelt sammankopplas via en av de två dedikerade Ethernet-MAC:ar som processorerna är utrustade med.

Som vanligt inom maskininlärning görs utveckling, träning och fintrimning på en arbetsstation som kanske också utnyttjar acceleratorer i molnet. När modellen tränats, trimmats och validerats färdigt, konverteras den för att kunna göra effektivare inferenser på en NPU-krets. Ett typiskt knep är att avrunda de flyttal på 32 bitar som träningen arbetade med, till heltal på 8 bitar.

Ytterligare sätt att effektivisera är att koppla in färdigbyggda nätskikt eller modeller redan optimerade för edge-tillämpningar.

Arcturus tillhandahåller en katalog med färdigbyggda modeller i olika precisioner. De är förvaliderade för att kunna användas i de viktigaste edge-körmiljöerna (Arm NN, TensorFlow Lite och TensorRT) i såväl CPU:er och GPU:er som NPU:er.

Här finns även verktyg för att träna och finjustera modeller liksom för att sortera datamängder och för att samla in och komplettera bilddata. Netto resulterar denna kombination av optimerad körtid, kvantiserad modell och NPU-hårdvara i en prestanda som är 40 gånger högre än när andra kända system kör samma modell.

Det är av avgörande betydelse att ett bibliotek av detta slag är komplett. Andra edge runtimes saknar ofta komplett stöd för alla typer av skikt i alla typer av nätverk. Nyare modeller med bättre prestanda tenderar att ha sämre stöd än de äldre typer som används i de välkända prestandamåtten.

Den sista komponenten som krävs är en inferensmotor med stöd för att ladda DNN-modeller i i.MX 8M Plus. NXP:s utvecklingsmiljö för maskininlärning, eIQ, innehåller porterade, validerade versioner av inferensmotorerna Arm NN och TensorFlow Lite.

Sammanfattning

Skalbar prestanda och flexibilitet är viktiga komponenter i maskininlärningstillämpningar i verkligheten. Varje applikation i sig är unik både vad gäller valet av DNN och den extra databehandling som krävs kring modellen. Det är avgörande att ha tillgång till ett ramverk som stödjer behovet av flexibilitet. Det är en viktig orsak till att kombinationen av den mikrotjänstarkitektur som utvecklats av Arcturus och den hårdvara för databehandling som utgörs av NXP i.MX8M Plus är ett kraftfullt verktyg när maskininlärning ska migreras till kanten. ■