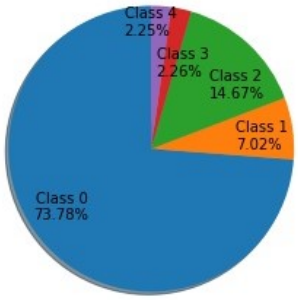
Retinaképek klasszifikálása konvolúciós neurális hálókkal

Önálló projekt II.
2021/22 2.félév

Vas Bernadett

Az önálló projektem második félévében a Kaggle EyePacs [4] retinafelvételeket tartalmazó adathalmazán végzett ötosztályos klasszifikáció problémáit, és azok megoldási lehetőségeit vizsgáltam. Az adathalmazon definiált fő feladat a diabetikus retinopátia betegség detektálása, illetve súlyosságának megállapítása volt. A retinafelvételek öt osztályba sorolhatóak annak alapján, mennyire előrehaladott a betegség a páciensnél:

- 0-ás osztály: teljesen egészséges
- 1-es osztály: enyhe
- 2-es osztály: közepesen súlyos
- 3-as osztály: súlyos
- 4-es osztály: poliferatív



1. ábra. A tanító adat eloszlása

Tehát a feladatunk egy adott képről megállapítani, hogy az melyik súlyossági stádiumba tartozik. Ehhez konvolúciós neurális hálókat használunk.

Több olyan tulajdonsággal is rendelkezik ez az adathalmaz, amit nem szabad figyelmen kívül hagyni. Az egyik ilyen a kiegyensúlyozatlanság: a tanítóhalmazban dominálnak az egészséges osztályba tartozó felvételek, míg a legsúlyosabbnak számító 3-as és 4-es osztályok nagyon alulreprezentáltak. Az adathalmaz eloszlását a 1. ábra mutatja. Az adathalmaz ezen eloszlása valójában nem meglepő, hiszen egy betegség természetszerűleg is úgy jelenik meg, hogy jóval kevesebb a betegségében előrehaladottabb állapotú páciens, mint az egészséges. Azonban mély hálók tanításánál problémát jelenthet az ilyen adateloszlás. Az algoritmusok túlságosan rá tudnak tanulni a domináns osztályra, ugyanis sok példát látnak belőlük, míg a kis létszámú osztályokból túl keveset ahhoz, hogy az jelentősebben befolyásolja a háló súlyainak beállítását, így nehezebben tanulják meg ezen képek felismerését. Orvosi alkalmazásoknál azonban az lenne a fontos, hogy a betegséget minél hamarabb felismerjük, azaz olyan algoritmusra van szükségünk, ami célszerű reprezentációt tud készíteni a képekből, és így helyesen osztályozza a betegség különböző stádiumait.

A modellek tanításánál figyelembe kell vennünk azt a tényt, hogy a címkék skálaszerűek, egy romló folyamatot írnak le. Ez azt is jelenti, hogy van egy távolság a különböző címkepárok között, és nem is tekinthető minden téves klasszifikálás ugyanakkora mértékű hibának. Például ha egy 2-es osztályba tartozó képet 3-asnak prediktál a modell, az kevésbé rossz, mintha 4-esnek osztályozná. A címkék jelentése miatt felmerül, hogy többosztályú klasszifikáció helyett más feladatot kell megoldanunk az adathalmazon. Az irodalomban *ordinal classification*-nek hívják azt a fajta osztályozást, amikor diszkrét címkéink vannak, de adott rajtuk egy rendezés. Ebbe a feladattípusba sorolhatóak az betegségek súlyosságának meghatározásán túl például filmek, könyvek, egyéb szolgáltatások minőségének rangsorolása, valamint emberekről készült fényképekből a kor megállapítása is.

A Kaggle EyePacs még most is az egyik legnagyobb publikusan elérhető diabetikus retinopátia adathalmaznak számít, azonban hátránya, hogy sok rossz minőségű, zajos felvételt, illetve félrecímkézett képet tartalmaz. A félreannotálás orvosi hibából ered, így azt feltételezzük, természetszerűleg inkább a szomszédos címkehatárokon történhettek a hibák, az állapotromlás fokozatossága miatt.

A méréseket EfficientNet-B0 és EfficientNet-B3 modellekkel végeztem, 128-as batch mérettel, SGD optimalizálóval és 0.001 learning rate-tel, amit a felére csökkentettem tanítás közben, ha 5 epoch-ig nem javult a validációs eredmény. Szinte minden esetben a nagyobb B3 modell jobban teljesített mint a B0, így ennek eredményeit prezentálom.

Metrikák

A kiegyensúlyozatlan eloszlás és skálaszerű címkék miatt megfelelő metrikákat kell választani a modellek teljesítményének méréséhez. A teszthalmaz eloszlása hasonlít a tanítóhalmazéhoz, így egy sima pontosság mérése félrevezetően jó eredményt mutathat, hiszen az egészséges osztályba tartozó felvételeket könnyebb megtanulni jól osztályozni, azonban a betegség különböző stádiumait is megfelelően kell tudni megkülönböztetni. Emiatt a választott metrikánk a macro-recall. A macro-recall a minden egyes osztályra külön kiszámolt recall értékeket átlagolja. Ez azért előnyös, mert minden osztályt ugyanakkora súllyal vesz figyelembe, tehát annak is nagy hatása lesz a végső eredményre, ha egy alulreprezentált osztályt rosszul klasszifikál a modell.

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \qquad Macro - Recall = \frac{\sum_{i=1}^{C} Recall_i}{C}$$

Ez a metrika azt veszi figyelembe, hogy pontosan ugyanaz-e a predikció mint a címke, azonban érdemes úgy is mérni a teljesítményt, hogy mennyire messze prediktál a háló a valódi címkétől. Erre a célra kvadratikus Cohen-kappa-t használunk. Emellett sokat elárul az osztályozás milyenségéről a tévesztés mátrix is.
A kvadratikus Cohen-kappa két értékelő közötti megegyezést mér, kvadratikusan súlyozva azokat az eseteket, ahol a két értékelő nem ért egyet. Most az egyik értékelő az orvosok által megadott valódi címkeeloszlás lesz, a másik a modellünk. A Cohen-kappa -1 és 1 között vesz fel értékeket, ahol 1 a teljes egyetértést jelenti, 0 a véletlen megegyezést, -1 pedig a teljes egyet nem értést. A metrika egy W súlymátrixból, O tévesztés mátrixból, és E mátrixból épül fel, ahol utóbbi a véletlen megegyezés valószínűségeit tartalmazza. Így a következő képlet adja meg nekünk a végső eredményt, ahol $N = 5$ az osztályok száma, i, j indexek pedig az i, illetve j osztályt jelölik:

$$\kappa = 1 - \frac{\sum_i \sum_j w_{i,j} o_{i,j}}{\sum_i \sum_j w_{i,j} e_{i,j}} \qquad w_{i,j} = \frac{(i-j)^2}{(N-1)^2}$$

Kiegyensúlyozatlanság vizsgálata

Először az EfficientNet modellt a teljes tanítóhalmazon tanítottam és lemértem a teljesítményt a teszthalmazon, amiben az osztályok aránya ugyanolyan volt, mint a tanítóhalmazban. Az eredmény Cohen-kappája 0.7020, macro-recall értéke 0.5029, a tévesztés mátrix pedig a 2a. ábrán látható. Ebből az olvasható le, hogy a 0-s osztályt szinte kiválóan tudja osztályozni, az 1-es osztály példányait viszont teljesen összetéveszti a 0-s osztályéval. Ennek oka az lehet, hogy a két osztály példányai túlságosan közel vannak egymáshoz, nehezen különböztetik meg a modellek a betegség tényleges első jeleit a képeken levő zajtól. Ugyanakkor az sem kizárható, hogy annotálási hibákkal állunk szemben, pontosan ugyanezen ok miatt. Meglepő, hogy a 2-es osztály felvételeinek egy részét helyesen osztályozta a modell, de 36%-ukat 0-nak prediktálta. Ezenkívül nagyobb hibák a 2-3 osztály határán fordulnak elő.

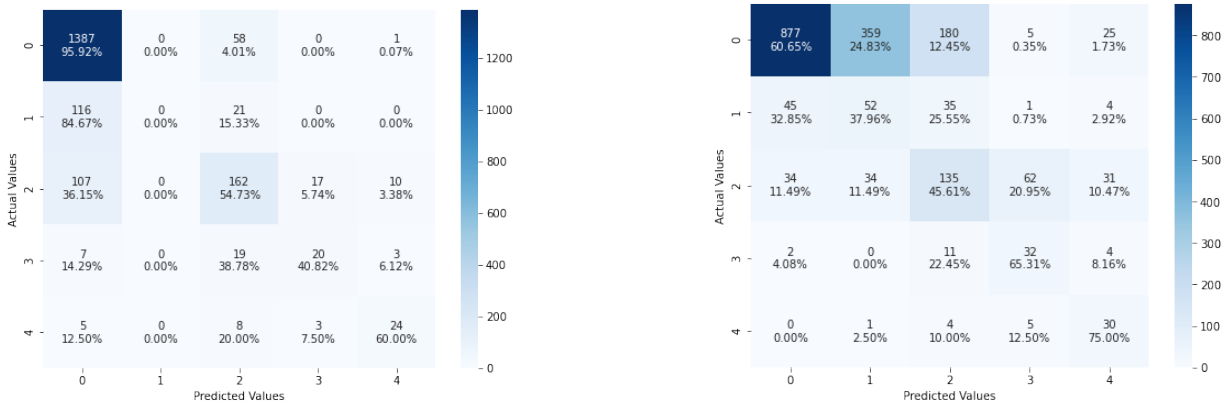
Az adathalmazban megjelenő kiegyensúlyozatlanságot több technikával is lehet kezelni [5]. Ezeket szétoszthatjuk adatszintű, illetve algoritmus szintű metódusokra. Adatszintű metóduson azt értjük, amikor az adat eloszlásának megváltoztatásával érjük el azt, hogy hasonló számban legyenek reprezentálva az osztályok. Ezzel szemben az algoritmus szintű technikáknál meghagyjuk a tanítóadat eredeti eloszlását, és a tanítási mechanizmust változtatjuk meg oly módon, hogy nagyobb figyelmet fordítsunk az alulreprezentált osztályokra. Mindkét fajta megközelítéssel végeztem méréseket az adathalmazon.

Először adatszintű átalakítást hajtottam végre, random oversampling-et és random undersampling módszert egyszerre alkalmazva. A 0-s osztály elemszámát visszatevés nélkül mintavételezve lecsökkentettem 6000-re, valamint a többi osztályból többszörösen véve minden adatpontot, azokat 6000 és 7000 közötti elemszámra növeltem fel. Azért láttam szükségesnek egyszerre végezni a két módszert, hogy egyrészt ne csökkentsük le jobban a 0-s osztály elemszámát, másrészt a 3-as, 4-es osztály minden

elemét ne kelljen túl sokszor bevenni. Utóbbi osztályok példáit ötször vettem be a tanítóhalmazba, és véletlenszerű augmentálást alkalmaztam rájuk. Az így kapott adathalmazon tanítva a modellt, a teszthalmazon 0.5646 macro-recall-t, illetve 0.57 cohen-kappát kaptam. A tévesztés mátrix a 2b. ábrán látható, és meg is magyarázza a kapott eredményeket. Az eloszlás megváltoztatása által elértük azt, hogy az 1-es osztályt jobban elkezdte felismerni a háló, de ugyanakkor a 0-s példákat jobban téveszti el. Javulásnak mondható még, hogy lecsökkent a 2-esnek prediktált 3-as képek aránya.

Használtam továbbá two-phase learning technikát az adaton [6], ami a tanítási folyamatot két fázisra osztja, az elsőben kiegyensúlyozott adaton tanítjuk a modellt konvergenciáig, majd az így megkapott súlyokkal tovább tanítjuk az eredeti eloszlású adathalmazon. Az előtanítás segíthet beállítani a neurális háló súlyait úgy, hogy az ne tanuljon rá túlzottan a domináns osztályra, majd a tanítás második fázisában transfer learning-gel ezek a súlyok tovább finomodhatnak az adathalmazban természetszerűleg jelenlevő eloszlásra. Eredményként 0.5283 macro-recall-t és 0.737 Cohen-kappa értéket kaptam, ami minimális javulást jelent az alapmodell eredményeihez képest. A tévesztés mátrixon látható (3a. ábra), hogy a 4-es és 0-s osztályoknál a modell az igazi címkéhez közelebb álló értékeket prediktált.

Az algoritmusszintű metódusok közül a veszteségfüggvény súlyozásának hatását vizsgáltam. Tanításhoz használhatunk olyan veszteségfüggvényt, ahol az adott osztályokhoz tartozó adatpontokat megsúlyozva elérhetjük, hogy bizonyos osztályokon elkövetett hibáknak nagyobb hatása legyen a súlyok változásaira. Háromféle súlyozást próbáltam ki a cross-entropy függvényen. Ezeknek annyi hátránya volt, hogy nem vették figyelembe a címkék között megjelenő rendezettséget,csak az adott osztályba tartozást. Elsőként a súlyok $w_i^{sq} = \frac{1}{\sqrt{c_i}}$ alakúak voltak, majd $w_i^t = \frac{total}{5 \cdot c_i}$, és végül $w_i^{en} = \frac{1-\beta}{1-\beta^{c_i}}$ ahol c_i az i-edik osztály elemszámát jelöli, a *total* a teljes tanítóhalmaz elemszámát, $\beta \in [0, 1)$ pedig egy hiperparaméter, jelen méréseknél $\beta = 0.9999$. A [3] cikk az egyes osztályok effektív példányainak számát alapul véve dolgozza ki a w_i^{en} súlyozási módszert. A kapott eredmények az 1. táblázatban láthatóak. A w_i^t és w_i^{sq} súlyozással az alap teljesítményhez képest javult a recall érték, azoban előbbinél jelentősen rosszabb kappa értékeket kaptunk. Ez azt vonja maga után, hogy bizonyos adatpontokat pontosabban prediktált a háló, amit viszont nem, azt a tényleges címkétől valamivel messzebbre jósolta.



(a) Modell alapteljesítménye az eredeti tanítóhalmazon

(b) Modell teljesítménye over- és undersampling módszerrel

2. ábra. Tévesztés mátrixok

Ordinal classification

Az ordinal classification standard megközelítése mellett frissebb munkák is foglalkoznak ezen témával és megoldásával. Cheng és tsai. írták le a neurális hálókra vonatkozó alapvető megközelítést, ezt NNRank-nek nevezték el [2]. Egy általános klasszifikációnál az $\mathbf{x}_i$ adatpontok címkéi one-hot vektorral vannak reprezentálva, azaz ha $\mathbf{x}_i$ a k-adik osztályba tartozik, akkor a ez a k-adik egységvektort jelenti, ahol a vektorok osztályszámnyi dimenziósak. A háló kimenete egy valószínűség eloszlás, tehát egy osztályszámnyi hosszú $\mathbf{o}$ vektor, aminek k-adik eleme a k-adik osztályba tartozás valószínűségét adja meg. A cél, hogy ha $\mathbf{x}_i$ címkéje k, akkor o_k közel legyen 1-hez, a többi elem pedig 0-hoz. Ezzel szemben, ha figyelembe akarjuk venni a címkék közti rendezést, akkor a one-hot vektorok helyett tekintsük úgy, hogy az adott adatpont minden nála kisebb osztályba is beletartozik. Tehát ha $\mathbf{x}_i$ címkéje k, akkor az

ehhez tartozó one-hot vektorban az $0 \leq i \leq k$ koordinátákon mind 1-es áll, a többi helyen 0. Nevezzük ezt a vektort a one-hot kiterjesztettjének. A háló $\mathbf{o}$, osztályszámnyi hosszú kimenetétől azt várjuk, hogy minden i indexre $o_i \in [0,1]$ és minden $j \leq k$ indexre o_j közel legyen 1-hez. Ekkor $\sum_{i=1}^{K} o_i$ nem 1 lesz, hanem egy becslés arra, melyik osztályba tartozik az $\mathbf{x}_i$ adatpont. A modell utolsó sűrű rétege után mind az 5 kimeneti neuronra sigmoid függvényt alkalmazva egy $[0,1]$ közti számot kapunk, ami annak a valószínűségét közelíti, hogy az adatpont minden mástól függetlenül a kimeneti neuronhoz tartozó osztályba való. Ezzel a megközelítéssel probléma, hogy a kapott o_i számok nem feltétlenül lesznek monotonak: $o_1 \geq o_2 \geq ... \geq o_K$. A modell által adott kimenetekből egy küszöbértékkel való összehasonlítással kapunk konkrét predikciókat. Az $\mathbf{o}$ vektor elejéről indulva végignézzük, hogy az egyes elemek nagyobbak-e mint a küszöbérték, ami lehet például 0.5. Ha igen, 1-est írunk a helyére, ha nem, 0-ást. A cél, hogy az így kapott vektor egyezzen meg a célváltozó kiterjesztett vektorával.

A leírt megközelítésben úgy is tekinthetünk a kimenetre, mint $K-1$ individuális bináris klasszifikátorra. Közülük a k-adik azt mondja meg, hogy az adatpont címkéje a k-adik súlyossági szintet megugorja-e. Ami probléma felmerülhet, hogy inkonzisztens predikciókat kapunk a bináris klasszfikátoroktól. Például ha a k-adik osztályzó azt adja ki, hogy az adott retina súlyosabb, mint a 3-as kategória, de egy ezt megelőző pedig 2-esnél enyhébbre jósolja. Az inkonzisztencia leküzdésére Raschka és tsai. írnak le egy technikát, ami bármilyen neurális hálóra átvihető [1]. A cikkben ResNet-34-el dolgoznak és emberek korát megjósló feladaton. Én az ötletet implementáltam az EfficientNet modellekhez.

A tanítás során az igazi címkéket ismét kiterjesztett one-hot vektorokkal reprezentáljuk. Egyetlen egy modell architektúra az alap, és a kimeneti sűrű réteget változtatjuk meg úgy, hogy $K-1$ darab f_i bináris klasszifikátort kapjunk, ahol K az osztályok száma. Így a modellből egy $K-1$ hosszú kimenetet kapunk, ahol minden elem $[0,1]$-beli, és összegezve kapjuk a kívánt predikciót. A monotonitást, azaz hogy $f_1(\mathbf{x}_i) \geq f_2(\mathbf{x}_i) \geq ... \geq f_{K-1}(\mathbf{x}_i)$ az garantálja, hogy mind a $K-1$ klasszifikátor ugyanazokkal a $\mathbf{w}$ súlyparaméterekkel rendelkezik, de más b bias értékekkel. Összegezve tehát, a modell kimenete a k-adik klasszifikátornál $\sigma(g(\mathbf{x}_i, \mathbf{W}) + b_k)$ ahol σ a sigmoid függvényt jelöli. A veszteségfüggvény a $K-1$ klasszifikátor cross-entropy-jának a súlyozott értéke:

$$L(\mathbf{W}, \mathbf{b}) = -\sum_{i=1}^{N} \sum_{k=1}^{K-1} \lambda^{(k)} \left[log(\sigma(g(\mathbf{x}_i, \mathbf{W}) + b_k)) y_i^{(k)} + log(1 - \sigma(g(\mathbf{x}_i, \mathbf{W}) + b_k))(1 - y_i^{(k)}) \right],$$

ahol $\lambda^{(k)}$ a k-adik feladat fontossági paramétere. Az adatpontokra a predikciókat hasonló módon kapjuk meg, mint a fent leírt esetben: $\sum_{k=1}^{K} f_k(\mathbf{x}_i)$ ahol $f_k(\mathbf{x}_i) = \mathbb{1}\{\sigma(g(\mathbf{x}_i, \mathbf{W}) + b_k) > 0.5\}$.

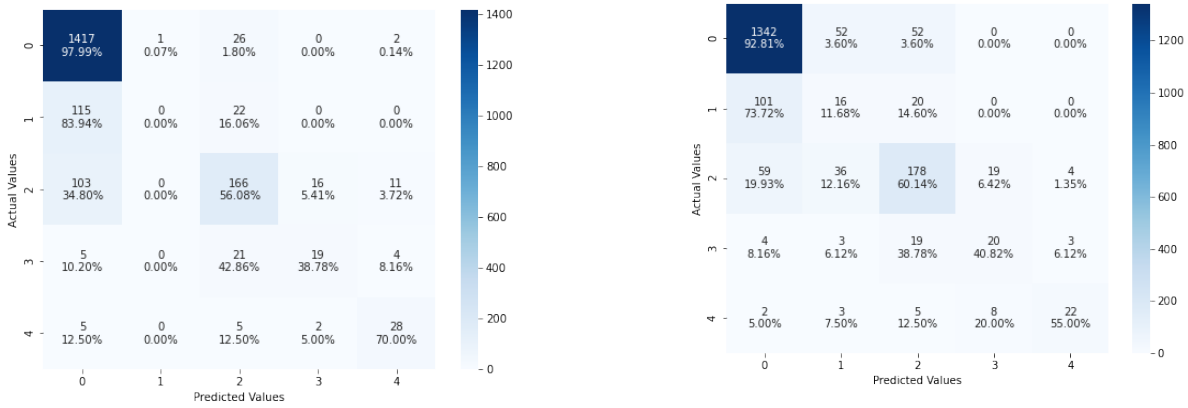
A fenti módszerrel tanítva az EfficientNet-B3-as modellt az 3b. ábrán látható tévesztés mátrixot kaptam a teszthalmazon, illetve 0.5208-as macro-recall és 0.7607- es Cohen-kappa értéket. Javulás az 1-es osztálynál mutatkozik meg, az osztály 11%-át prediktálta jól az alapmodell 0%-ához képest. A 2-es osztálynál több pontos, és az igazi értékhez közelebbi találat lett, utóbbi állítás igaz a 3-as és 4-es osztályra is.

	Macro-recall	Cohen-kappa
Alap teljesítmény	0.5029	0.7020
Oversampling + Undersampling	0.5646	0.5700
Two-phase learning	0.5283	0.737
Loss súlyok: w_i^{sq}	0.5508	0.7173
Loss súlyok: w_i^{t}	0.5578	0.6238
Loss súlyok: w_i^{en}	0.5377	0.7095
Kiterjesztett one-hot vektoros megközelítés	0.5208	0.7607

1. táblázat. Eredmények

Konklúzió és további tervek

A mérésekből megtapasztalható, hogy a feladat nehezen tanulható, legfőképp az általunk vizsgált szemszögből, azaz hogy minél pontosabban felismerjük a betegség stádiumait. A tévesztés mátrixokból az látszik, hogy a legnehezebben megkülönböztethető osztálypárok a 0-1, illetve 2-3. Az under- és oversampling esetében bár megváltoztattuk az adat természetes eloszlását, de ezzel rábírtuk a modellt arra, hogy jobban felismerje az 1-es, illetve 3-as osztály adatpontjait. A kipróbált módszerekkel



(a) Modell teljesítménye two-phase learning-gel

(b) Modell teljesítménye kiterjesztett címkékkel

3. ábra. Tévesztés mátrixok

tanított hálók különböző mértékben ragadtak meg jellemzőket az egyes osztályokból, így természetesen adódik a felvetés, hogy érdemes lenne ensemble modellt építeni a feladatra. Ez többféleképpen is megvalósítható. Történhet a fenti módszerekkel tanított modellek összerakása és predikcióik okos kiátlagolásaként, vagy akár kisebb feladatokon való tanításként. Erre egy példa, hogy építhetünk modellt a nehezen megkülönböztethető osztályokra, tehát hogy 0-1, illetve 2-3 osztályok között végezzünk bináris klasszifikációt.

Az adathalmaz kiegyelítetlenségének problémáját célzó megoldásokat érdemes lenne az ordinal classification feladatba beágyazni. Továbbá, a veszteségfüggvény súlyozáson lehetne finomítani azzal, ha különbséget tennénk az egyes hibázások között. Nem ugyanolyan súlyos hibának számít az, hogy ha egy szomszédosra prediktáljuk félre az igazi címkét, vagy ha egy sokkal távolabb levőre. Orvosi feladatként még szimmetriát sem feltétlen tételezhetünk fel téves klasszifikálásnál, például nem mindegy, hogy 2-es súlyosságot állapítunk meg egy 4-es stádiumú retinára, vagy fordítva. Ezenkívül a képek alaposabb előfeldolgozása, minőségük javítása is hozzájárulhat a hálók teljesítménynövekedéséhez.

Hivatkozások

[1] Wenzhi Cao, Vahid Mirjalili, and Sebastian Raschka. Rank consistent ordinal regression for neural networks with application to age estimation. *arXiv preprint arXiv:1901.07884*, 2019.

[2] Jianlin Cheng, Zheng Wang, and Gianluca Pollastri. A neural network approach to ordinal regression. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pages 1279–1284. IEEE, 2008.

[3] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019.

[4] Diabetic Retinopathy Detection Dataset. https://www.kaggle.com/c/diabetic-retinopathy-detection/data.

[5] Justin M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. 2019.

[6] Hansang Lee, Minseok Park, and Junmo Kim. Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. In *2016 IEEE international conference on image processing (ICIP)*, pages 3713–3717. IEEE, 2016.